

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Sistema de acompanhamento curricular

Jorge Filipe Vieira Barbosa Teixeira



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Raul Fernando de Almeida Moreira Vidal

Coorientador: João Carlos Pascoal Faria

23 de Julho de 2017

Sistema de acompanhamento curricular

Jorge Filipe Vieira Barbosa Teixeira

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: António Miguel Pontes Pimenta Monteiro

Arguente: Paulo Sousa

Orientador: Raul Fernando de Almeida Moreira Vidal

23 de Julho de 2017

Resumo

Neste momento, o contexto atual do acompanhamento curricular do Departamento de Engenharia Informática da FEUP, contempla vários órgãos e intervenientes que têm diferentes papéis e responsabilidades nos vários cursos que o departamento tem em si inseridos. Para além do grande número de pessoas que estes grupos contêm, a grande variedade de formas de comunicação existentes no momento - comunicação oral, e-mails, reuniões pedagógicas ou até o uso de redes sociais - aumenta em demasia a complexidade do acompanhamento curricular neste sistema atual.

Todos estes fatores tornam a tarefa de gerir o acompanhamento curricular bastante complexa e abrem a possibilidade de existirem falhas no tratamento de algumas situações.

Sendo assim a proposta desta dissertação é que seja criada uma plataforma capaz de sintetizar algumas destas responsabilidades num sistema global, onde os vários atores - quer sejam professores, alunos ou comissões - possam otimizar o processo de resolução de issues que possam ser identificados no decorrer das tarefas curriculares.

Desta forma, o sistema proposto deve incorporar um Issue Tracking System aplicado ao acompanhamento curricular, numa primeira fase virado para o Departamento de Engenharia Informática da FEUP, mas com o propósito de ser expandido para toda a comunidade da FEUP.

O sistema está dividido por anos e unidades curriculares e cada uma destas divisões poderá ser acedida por diferentes utilizadores, dependendo do nível de permissões que cada utilizador tiver. Um administrador tem acesso a todas as unidades curriculares do curso e a um campo de estatísticas concretas do estado atual de cada ano e de cada unidade curricular. Este estado é discriminado pelo número de issues aberto em cada momento e pode ser regulado pelo administrador. Um encarregado de uma unidade curricular (professores dessa UC, ou um aluno responsável pelo ano), podem ter acesso aos issues desta unidade curricular e podem contribuir para a resolução desse issue na página do mesmo. Um aluno tem apenas acesso às unidades curriculares que frequenta e pode ver o histórico de issues abertos e criar novos issues.

Os issues devem ser catalogados como “Problema”, “Proposta”, ou “Sugestão”, de forma a poderem ser analisados de forma mais correta e tornar o processo de resolução mais célere.

Nesta fase inicial é esperado que o sistema seja posto à prova com uma utilização real, tomada a cabo pelos elementos constituintes da comissão de acompanhamento do MIEIC, que servirá de teste para o sistema, para que no futuro este possa ser alargado a toda a comunidade da FEUP.

Deverão também ser incluídas algumas funcionalidades extra tais como a criação de grupos de intervenientes por unidades curriculares - onde alunos inscritos a uma UC podem ter acesso a informação disponibilizada pelos professores dessa UC, ou pelo secretariado do DEI, evitando assim o envio de e-mails com este fim - ou grupos de alunos de anos curriculares, onde estes possam interagir e partilhar informações das suas UC, criando uma alternativa aos grupos em redes sociais existentes atualmente.

Para além das funcionalidades aqui apresentadas, o sistema poderá acolher funcionalidades complementares em trabalhos futuros.

Abstract

At this moment, the current context of the curricular monitoring of the Department of Informatics Engineering of FEUP, contemplates several groups of intervenients which have different roles and responsibilities in the various courses that the department has inserted in itself.

For in addition to the large number of people these groups contain, the wide variety of forms of communication - oral communication, e-mails, meetings or even the use of social networks - greatly increases the complexity of the curricular monitoring in the current system.

All these factors makes the task of managing the curricular follow up quite complex and open the possibility of mistakes in the treatment of some situations.

Therefore, the proposal of this dissertation is to create a platform capable of synthesizing some of these responsibilities in a global system, where the various actors - whether teachers, students or groups- can optimize the problem solving process that can be identified during the academic period.

In this way, the proposed system should incorporate an Issue Tracking System applied to the curricular follow-up, in a first phase turned to the Department of Informatics Engineering of FEUP, but with the purpose of being expanded for the entire FEUP community.

The system is divided by years and courses and each of these divisions can be accessed by different users, depending on the level of permissions that each user has. An administrator has access to all courses and to a set of statistics of the current state of each year and each course. This state is defined by the number of issues open at any time and can be regulated by the administrator. A person who manage a course (teachers of this course, or a student representative of the year), can have access to the issues of this course and can contribute to the resolution of this issue in its page. A student has only access to the courses that he / she attends and can see the history of open issues and create new issues.

Issues should be categorized as "Problem", "Proposal", or "Suggestion", so that they can be analyzed more accurately and make the resolution process faster.

At this early stage, it is expected that the system will be put to the test with a real use made by the constituent elements of the MIEIC affair council, which will serve as a test for the system, so that in the future it can be extended to the whole community of FEUP.

Some extra features should also be included, such as the creation of groups by course - where students enrolled in a course may have access to information provided by the course professors, or by DEI secretariat, thus avoiding sending e-mails to this ends - or groups for students enrolled in the same academic year where they can interact and share information from their courses, creating an alternative to currently existing social networking groups.

In addition to the features presented here, the system will be able to host complementary functionalities in future works.

Agradecimentos

A realização de qualquer dissertação pressupõe um nível de esforço, empenho e dedicação incomparável. No entanto, só desta forma seria justo concluir um curso de mestrado, tão exigente em todos os momentos como é o caso do Mestrado em Engenharia Informática e Computação da FEUP. É sem dúvida, com grande felicidade e sentido de dever cumprido que chego a esta fase do meu percurso académico. Mas nada disto seria possível sem um conjunto de pessoas que tornaram o meu percurso mais fácil e fizeram com que tudo valesse a pena:

Primeiro, quero agradecer ao meu orientador, prof. Raúl Vidal por todo o apoio que deu ao desenvolvimento deste projeto. Numa nota paralela, desejo que nesta fase, que também é de mudança e conclusão para o professor, que tudo lhe corra pelo melhor e que se mantenha próximo dos alunos do nosso curso, porque a sua experiência é ainda necessária.

Agradeço também ao prof. Pascoal Faria, meu coorientador, pelo acompanhamento que dedicou a este trabalho e também no resto do meu percurso, que foi sem dúvida dos mais importantes.

Quero agradecer a todos os colegas e amigos que ganhei durante o meu percurso na FEUP, porque eles foram fundamentais para que a vontade por continuar nunca faltasse.

Dentro destes amigos, preciso de fazer um agradecimento particular à amiga de todos os momentos que me acompanhou, encaminhou e aturou durante os últimos cinco anos e espero que por muitos por vir.

Para finalizar, tenho que fazer o maior agradecimento de todos à minha família, em particular aos meus pais e irmã que sempre me aconselharam e deram força, confiança e apoio para que pudesse continuar o meu trabalho e o meu percurso.

Jorge Filipe Teixeira

*“Now this is not the end.
It is not even the beginning of the end.
But it is, perhaps, the end of the beginning”*

Winston Churchill

Conteúdo

1	Introdução	1
1.1	Contexto do problema	2
1.2	Motivação e objetivos	2
1.3	Estrutura da Dissertação	3
2	Sistemas de gestão de aprendizagem e de projetos	5
2.1	Learning Management System	5
2.2	<i>Issue Tracking System</i>	9
2.3	Análise das tecnologias existentes	13
2.4	Conclusões	14
3	Sistema de Informação para o acompanhamento curricular	15
3.1	Modelo atual utilizado na FEUP	15
3.2	Problemas do modelo atual	16
3.3	Integração de um sistema de informação para o acompanhamento curricular . . .	17
3.3.1	Principais Funcionalidades	17
3.3.2	Mais valias do sistema	19
3.3.3	Métodos de validação do sistema	20
3.3.4	Melhoramentos ao sistema proposto	20
4	Solução Desenvolvida	23
4.1	Arquitetura do sistema	23
4.2	Tecnologias	24
4.3	Plataforma	25
4.3.1	Servidor	25
4.3.2	Cliente	26
4.3.3	Estrutura de dados	26
4.3.4	API	28
4.4	Funcionalidades implementadas	30
4.5	Fluxo de utilização	31
5	Testes e resultados obtidos	41
5.1	Metodologia dos testes realizados	41
5.2	Resultados e observações	44
5.2.1	Resultados referentes à comparação entre o sistema atual e o proposto . .	44
5.2.2	Resultados referentes à avaliação das funcionalidades desenvolvidas . . .	45
5.3	Conclusões	46
6	Conclusões	49

CONTEÚDO

Referências	53
A Resultados obtidos nos testes realizados	55

Lista de Figuras

2.1	Bugzilla Issue List	11
2.2	Jira project overview	12
2.3	Github Issue Listing	13
4.1	Arquitetura do sistema	24
4.2	Página inicial	32
4.3	Utilizador não validado	33
4.4	Estudante validado	33
4.5	Gestor validado	34
4.6	<i>Dashboard</i> de administrador	35
4.7	Criação de um novo grupo	35
4.8	Página de estatísticas do sistema	36
4.9	Listagem de utilizadores do sistema	36
4.10	Listagem de problemas do sistema	37
4.11	Página de um grupo - Administrador	37
4.12	Página de um grupo - Estudante	38
4.13	Alterar permissões de um utilizador	38
4.14	Página de um problema - Estudante	39
4.15	Página de um problema - Administrador	39
A.1	Resultados obtidos pelo teste comparativo entre sistemas	56
A.2	Resultados obtidos pelo teste às avaliações do sistema proposto	57

LISTA DE FIGURAS

Lista de Tabelas

2.1	Critérios a avaliar sobre <i>learning management systems</i> - Parte 1	7
2.2	Critérios a avaliar sobre <i>learning management systems</i> - Parte 2	8
2.3	Resultados obtidos sobre avaliação aos <i>learning management systems</i>	8

LISTA DE TABELAS

Abreviaturas e Símbolos

DEI	Departamento de Engenharia Informática
FEUP	Faculdade de Engenharia da Universidade do Porto
ITS	Issue Tracking System
LMS	Learning Management System
UC	Unidade Curricular

Capítulo 1

Introdução

Numa organização que contemple um vasto número de intervenientes, em que estes tenham papéis ativos diferenciados e com responsabilidades distintas, é de grande importância que alguns mecanismos de assistência e acompanhamento ao normal funcionamento das funções da organização, sejam implementados e preservados.

O papel deste tipo de acompanhamento numa organização de ensino é essencial para o natural decorrer do desenvolvimento académico quer dos alunos, quer dos professores, passando também pela melhoria do funcionamento administrativo destas organizações. Este tipo de acompanhamento é normalmente da responsabilidade de um conjunto de elementos da organização, que têm a tarefa de manter o bom funcionamento das atividades das instituições.

Grande parte das instituições da área, adotam atualmente um *Learning Management System* (LMS), sendo o mais comum destes sistemas o *Moodle*. Os números da ferramenta *Moodle* demonstram a importância deste sistema nas várias organizações em todo o mundo, existindo 74,744 *sites* registados num total de 231 países. Ao todo este sistema chega a mais de 98 milhões de utilizadores registados¹.

Um dos pontos fortes a favor da ferramenta *Moodle* é a versatilidade que se pode imputar ao sistema, de forma a servir o máximo de propósitos possível naquilo que compete à comunicação, avaliação e publicação de temas a nível académico. Algumas destas funcionalidades passam por disponibilizar conteúdos multimédia, eventos, notícias, fóruns, questionários, inquéritos, *chat*, correio eletrónico, *wiki*, etc. A comunidade *open-source* disponibiliza vários modelos adicionais, produzidos por diferentes pessoas, que podem ser adicionados ao *Moodle* [KMAC08].

Apesar de todos estes fatores referidos, no que toca ao acompanhamento académico ainda ficam em falta vários casos de uso que seriam uma mais valia para a resolução de problemas nesta área e também não são incluídos nestes sistemas todos os intervenientes ativos nestas organizações de ensino.

¹ *Moodle*; <https://moodle.net/stats/>; 2017

1.1 Contexto do problema

Atualmente na Faculdade de Engenharia da Universidade do Porto, o acompanhamento académico é assegurado por um órgão interno da faculdade, dividido pelos vários departamentos que constituem a Faculdade e que asseguram que os interesses dos alunos e professores sejam mantidos, para além de garantirem que o funcionamento das atividades dos vários departamentos decorram com normalidade, são estas as comissões de acompanhamento.

Na Faculdade de Engenharia, o trabalho destas comissões de acompanhamento passa por auxiliar a comunicação aluno-professor, professor-aluno e com o departamento em questão, alargando também este contacto aos auxiliares administrativos, facilitando assim a interação interna da organização e fomentando o trabalho em conjunto dos vários intervenientes com parte ativa no quotidiano da faculdade. Para além disto, é de grande importância que a comissão de acompanhamento consiga garantir que os vários problemas que vão sendo levantados ao longo do tempo, que passaremos a referir como *issues*, possam ser resolvidos prontamente.

Apesar do trabalho realizado por estas comissões, é essencial aumentar o alcance que este acompanhamento traz às instituições de ensino, de forma a ser possível manter um acompanhamento mais próximo das situações que exijam uma atenção maior em cada momento e para promover um funcionamento mais rápido e direto em situações de necessidade imediata. Até porque este acompanhamento se torna uma tarefa exigente, tendo em conta a dimensão do universo de intervenientes que as comissões se propõem a cobrir (alunos, docentes, secretariado, órgãos de gestão, etc.), o escopo de assuntos que devem ser constantemente monitorizados (plano de estudos, métodos de ensino e avaliação, recursos físicos, etc.) e também a diversidade de meios de comunicação que os vários intervenientes adotam (e-mails, reuniões, inquéritos, relatórios e meios digitais como o *skype* ou as redes sociais).

1.2 Motivação e objetivos

Devido à importância crescente que este tipo de acompanhamento vai tendo nas instituições de ensino e pelo reconhecimento, que vai agregando por meio dos trabalhos realizados ao longo do tempo, é essencial que seja implementado um sistema *online*, capaz de aumentar o alcance do acompanhamento curricular e tornar mais rápido, de fácil acesso e resolução o tratamento dos vários problemas e situações que vão sendo identificados no decorrer do funcionamento normal das atividades das instituições.

Um sistema *online* que cumpra as exigências necessárias e que seja competente no cumprimento das tarefas propostas, deve englobar alunos, professores e auxiliares administrativos, dando assim uma visão geral do estado corrente das organizações. Para este sistema beneficiar de uma fácil e rápida leitura das situações mais pertinentes em cada momento, cada ator deverá ter uma vista personalizada com os assuntos do seu interesse e das suas competências.

Introdução

Uma das funcionalidades mais úteis neste tipo de sistema, deverá ser a possibilidade de consultar os históricos dos *issues* reportados até à data e as soluções adotadas para solucionar estes problemas.

Para que os diferentes tipos de intervenientes possam colaborar no sistema de acompanhamento, é importante que possa existir uma comunicação direta na resolução de *issues*, numa perspetiva de permitir uma maior rapidez de resposta aos diferentes problemas e para que todos se sintam integrados no próprio processo de resolução das ocorrências.

Por outro lado, de forma a simplificar a leitura e o tratamento destes mesmos problemas, será importante que o sistema possa conter subdivisões, de forma a agrupar os vários intervenientes pelas áreas que melhor podem servir, por exemplo divisões por anos curriculares (no caso dos alunos), ou por unidades curriculares (no caso dos professores).

1.3 Estrutura da Dissertação

A presente dissertação, para além da introdução, é composta por 4 capítulos que passaremos a enunciar:

No capítulo 2, é descrito o estado da arte e são apresentadas soluções de sistemas de informação que ajudam a perceber tanto o problema em que este trabalho se insere, como ajuda a perceber melhor que tipo de abordagens podem ser tomadas.

No capítulo 3, é apresentado o modelo utilizado atualmente na Faculdade de Engenharia da Universidade do Porto no que toca ao acompanhamento curricular, são destacados alguns problemas que devem ser melhorados e são posteriormente assinaladas as principais funcionalidades que o trabalho que é aqui apresentado deve conter.

No capítulo 4 é referida a arquitetura do sistema desenvolvido na fase de implementação, são explanadas as tecnologias que foram utilizadas para a realização deste sistema, são mostradas as funcionalidades implementadas e são explicados os tipos de utilização que podem ser feitos através desta plataforma.

No capítulo 5 podemos ver que tipos de testes de validação foram realizados e quais foram os resultados obtidos. Será feito também, um levantamento das conclusões e observações obtidas com estes testes.

Por fim, no capítulo 6 são referidas as conclusões que podem ser retiradas do trabalho de dissertação e quais são os planos de futuro para o trabalho que foi realizado.

Introdução

Capítulo 2

Sistemas de gestão de aprendizagem e de projetos

O desenvolvimento deste trabalho tem como um dos principais objetivos o processamento, análise e resolução de problemas, neste caso em concreto, aplicado às instituições de ensino com um foco extra na Faculdade de Engenharia da Universidade do Porto.

Neste capítulo serão apresentados alguns exemplos de sistemas e metodologias adotados para problemas com finalidades diferentes da que este trabalho se propõe a resolver, mas com alguns conceitos que podem ser úteis para construir uma proposta de resolução, que possa colmatar as lacunas que existem atualmente no panorama atual deste trabalho.

2.1 Learning Management System

O meio eletrónico de comunicação com alunos oferece várias vantagens que podem ser atingidas com soluções variadas. Entre elas, a mais popular é através do uso de *Learning Management System*[\[KMAC08\]](#).

Cada vez se torna mais importante para qualquer instituição de ensino, fazer-se servir de meios tecnológicos para fazer divulgação de materiais letivos para os alunos. Esta forma de proceder à partilha e publicação de conteúdos tem como grande vantagem, a possibilidade de a qualquer momento e em qualquer local o aluno ter acesso a esta informação, tornando o ensino mais versátil e assíncrono.

Outro benefício importante que advém do uso deste tipo de sistemas é o facto de ser um sistema intuitivo e que auxilia os seus utilizadores nas tarefas que tiverem que realizar. Facilmente os conteúdos podem ser adicionados, modificados, removidos, partilhados e reutilizados facilmente sem grandes esforços ou custos adicionais.

Sendo assim, um docente de uma instituição de ensino que se faça servir de um sistema deste género, tem a possibilidade de em qualquer momento e em qualquer lugar, disponibilizar conteúdos letivos, podendo estes ser do tipo escrito, digital ou multimédia por exemplo, para um conjunto de utilizadores do mesmo sistema.

Do ponto de vista dos alunos, estes podem ver conteúdos multimédia, comunicar com os seus professores e entre si em comunidades de aprendizagem, fazer o *download* de conteúdos letivos, responder a *quizzes* e submeter projetos e trabalhos de casa [KMAC08].

Outra possibilidade que estes sistemas permitem, é a criação de vários sub-grupos, divisões estas que podem ser feitas de forma a estruturar da melhor forma possível os utilizadores por conteúdos. Sendo assim, numa faculdade podem existir divisões por curso e por unidades curriculares, fazendo com que só os utilizadores associados a uma determinada unidade curricular possam ter acesso aos conteúdos que forem sendo disponibilizados e aos fóruns que estes subgrupos possam ter. Desta forma é possível manter os utilizadores de cada grupo apenas no mínimo necessário.

A necessidade de utilização deste tipo de sistemas pelas organizações de ensino, advém em grande parte da dificuldade que pode existir numa instituição em que, por não existir um sistema centralizado, a distribuição e partilha de conteúdos letivos, quer seja de docentes para alunos ou no sentido inverso, se pode tornar um processo bastante confuso e disperso [KMAC08].

No que toca à selecção de um LMS para uma instituição de ensino, podem ser tomadas decisões tendo em conta critérios de seleção diferentes. Os principais decisores poderão ser a usabilidade, o nível de confiança do sistema e o suporte [KPTD11]. No que toca aos critérios secundários de decisão, para o centro de investigação da Universidade Estatal do Idaho nos Estados Unidos, estes devem ter em conta o valor pedagógico, fatores económicos, integração com os serviços tecnológicos existentes e viabilidade a longo prazo .

Analizando os dados resultantes de um estudo comparativo entre vários LMS, usando numa primeira fase como critérios, o número de diferentes ferramentas e serviços que cada sistema oferece (e.g. ferramentas de apoio às cadeiras, ferramentas de criação de conteúdos, ferramentas de comunicação, ferramentas de assistência e ferramentas administrativas). E numa segunda fase foi analisada a adaptabilidade, o custo, a expansibilidade e a interoperabilidade de cada um [KPTD11]. Este estudo foi aplicado em nove sistemas LMS (ATutor¹, Blackboard²; Claroline³, Dokeos⁴, eCollege⁵, FLE, SAKAI⁶, ILIAS⁷ e Moodle⁸).

Depois de terem sido definidos os critérios a avaliar neste estudo comparativo, foi elaborada uma tabela, como é ilustrado nas imagens 2.1 e 2.2, onde estão referidos os pesos que cada funcionalidade oferecida tem no sistema a escolher, que em conjunto completam o primeiro estudo comparativo. No segundo estudo comparativo, são analisados os critérios secundários, referidos

¹ ATutor; <http://www.atutor.ca/>; 2017

² Blackboard; <http://uki.blackboard.com/>; 2017

³ Claroline; <http://www.claroline.net/>; 2017

⁴ Dokeos; <http://www.dokeos.com/>; 2017

⁵ eCollege; <http://www.ecollege.ie/>; 2017

⁶ SAKAI; <https://sakaiproject.org/>; 2017

⁷ ILIAS; <http://www.ilias.de/>; 2017

⁸ Moodle; <https://moodle.org/>; 2017

anteriormente. Sendo assim, cada grupo de funcionalidades tem uma determinada importância para a instituição e depois de realizar esta análise para todos os sistemas LMS enumerados anteriormente, aquele que obtiver um resultado mais alto, deverá ser o escolhido pela instituição. Os resultados obtidos estão representados na imagem 2.3 e mostram que o sistema LMS que obtém a melhor classificação é o *Moodle*.

Tools and Services	Significance
Communication tools	
Forum	2
Forum management	2
File transfer (FTP)	2
E-mail	2
Online journal /Notes	1
Sharing of online journal /Sharing of notes	2
Chat	1
Whiteboard	2
Teleconference	1
Assistance tools	
Bookmarks	1
Search of educational content	2
Calendar	2
Online help for the students	2
Online help for the tutors	2
Group specific spaces	2
Self-evaluation tools	2
Creation of groups of students	2
Portfolio	1
Administration tools	
User authentication	2

Tabela 2.1: Critérios a avaliar sobre *learning management systems* - Parte 1

Tools and Services	Significance
Role management	2
Students' enrolment in various courses	2
Course tools	
Scheduling of educational activities	1
Grading tools	2
Students' tracking tools	2
Test creation	2
Course management	2
Educational content creation tools	
Course templates	2
Graphical user interface (GUI) customization	2
Courses organization	2
Instructional design tools	2
Case-based learning tools	
Support of cases for instruction	2
Support of cases for examination	2
Application in different field	2
Support during learning process	2
Customization	2
Local installation	1

Tabela 2.2: Critérios a avaliar sobre *learning management systems* - Parte 2

Evaluation Criteria	ATutor 1.5.4	Blackboard Vista 4.1 Enterprise License	Claroline 1.8.1	Dokeos 1.8	eCollege	FLE 3 1.4.2	SAKAI 2.3	ILIAS 3.7.7	Moodle 1.8
Results of the first benchmark	50	50	37	48	50	18	52	41	48
Adaptability	5	0	5	5	0	5	5	5	5
Cost vs. Open source	5	0	5	5	0	5	5	5	5
Project team experience	0	3	0	0	0	0	0	0	5
Expandability	5	0	4	5	0	3	5	5	5
Interoperability	5	2	4	4	2	3	5	5	5
Total of benchmarks 1 & 2	70	55	55	67	52	34	72	61	73

Tabela 2.3: Resultados obtidos sobre avaliação aos *learning management systems*

Moodle

O *Moodle* permite que os docentes possam avaliar trabalhos, disponibilizar aulas, partilhar documentos, criar questionários, disponibiliza *chats* e fóruns que são fáceis de utilizar e oferecem alta qualidade [AAZ08]. Esta é uma das ferramentas mais *user-friendly* e flexível daquelas que

são de carácter *open-source* e é projetada essencialmente para os docentes que pretendem criar um sistema *online* de grande qualidade. O sistema tem uma vasta documentação, um forte suporte para garantir a segurança e auxiliar a administração do sistema.

De forma a abranger tanto campos colaborativos como campos letivos, o *Moodle* é composto por módulos independentes, *plug-ins* [KGD11].

De acordo com Kumar *et al.* estes módulos podem ser divididos por diferentes grupos de interesse, módulos de comunicação, produtividade, relacionamento com os alunos, administrativo, gestão de cadeiras e gestão de plano curricular.

No que toca ao módulo de comunicação, as funcionalidades fundamentais são as que permitem troca de ficheiros entre utilizadores, fórum (interno ou externo) de discussão através de *e-mails* onde podem ser partilhadas imagens, *urls* ou vários tipos de anexos e *chats* em tempo real. O módulo de produtividade inclui ferramentas como calendários, pesquisas ou assistência *online*. No módulo de relacionamento com os alunos, os pontos mais importantes são a existência de *workshops*, possibilidade de elaborar trabalhos de grupo, desenvolver portfólios pessoais e realizar auto avaliações. No que toca ao módulo de administração as funcionalidades principais passam por autenticar membros, dar autorizações a membros que queiram pertencer a um grupo ou cadeira ou criação de registos do sistema. Quanto ao módulo de gestão de cadeiras, as funcionalidades principais são as de determinação de classificações *online*, gestão de plano das cadeiras, percurso dos alunos. Por fim, as funcionalidades que estão presentes no módulo de gestão de plano curricular são a possibilidade de customização e aceder a *templates* já existentes [KGD11].

Uma das principais vantagens do *Moodle* é o facto da comunidade estar em constante crescimento à volta do projeto. Tanto os *developers* como os utilizadores participam ativamente nos fóruns de discussão do *Moodle*, partilhando sugestões, ou partes de código, ajudando novos utilizadores, partilhando conhecimento e debatendo novas ideias [AAZ08].

2.2 Issue Tracking System

O *Issue Tracking System* (ITS) é um repositório que agrega todas as tarefas de desenvolvimento que um projeto deve conter (e.g. novas funcionalidades, resolução de *bugs*, etc.). Para cada uma destas tarefas, o ITS atribui uma descrição, informação de gestão (e.g. estado do *issue*), uma prioridade, comentários e anexos adicionados por *developers* de forma a discutir a tarefa em questão. Para além disto, vários *issues* podem ser relacionados uns com os outros, por exemplo é importante relacionar *issues* que estejam duplicados, de forma a que um seja concluído quando o outro estiver nesse estado [ODA⁺15].

Os ITS são bastante conhecidos na área de desenvolvimento de *software* e *IT support* [LMK15] e são cada vez mais utilizados, pois contribuem para que os *developers* possam ter um controlo maior sobre os seus projetos e sobre o trabalho que ainda deve ser realizado.

Este sistema ficou mais reconhecido por servir de base para a gestão de *bugs* de projetos de *software*. Desta forma os *developers*, à medida que iam descobrindo problemas no seu código que

devessem ser corrigidos, reportavam um *bug* e posteriormente esta lista de *bugs* que ia sendo construída, deveria ser solucionada de forma a poderem dar o trabalho como concluído. No entanto, o surgimento do termo *issue tracker*, em lugar do termo antecessor *bug tracker*, foi implementado na sequência da crescente percepção de que este tipo de sistema pode desempenhar um papel mais importante “do que simplesmente adicionar mais uma *to do list*” ao trabalho desempenhado pelos *developers* [JD03].

Sendo assim, utilizar um ITS significa um melhoramento de comunicação entre os membros de uma equipa. Para além disso, se o projeto em questão tiver uma dimensão significativa, este sistema pode também notificar utilizadores externos ao trabalho em questão, do desenvolvimento que tenha sido feito no que diz respeito a um pedido ou preocupação que tenha sido levantada por este à equipa de desenvolvimento [JD03].

Um ITS, por norma, tem várias funcionalidades que ajudam a equipa de trabalho a desempenhar melhor as suas funções, dando-lhes percepção do avanço do seu trabalho e do estado corrente dos problemas, sugestões ou notas que foram sendo observadas ao longo do avanço de um projeto. Estas funcionalidades passam por [JD03]:

- Classificação de utilizadores com base nos seus privilégios;
- Criação de novos *issues* pelos utilizadores, adicionando ficheiros relevantes;
- Classificação de *issues* por ordem de prioridades e áreas de interesse;
- Modificação de *issues*, podendo adicionar novos ficheiros;
- Pesquisa de *issues*, utilizando pesquisa por palavras ou termos, prioridade ou áreas;
- Procurar nas listas de *issues*, utilizando filtros;
- Comunicar com outros utilizadores sobre *issues* já reportados;
- Enviar e-mails com notificações referentes a modificações efetuadas em *issues* existentes;

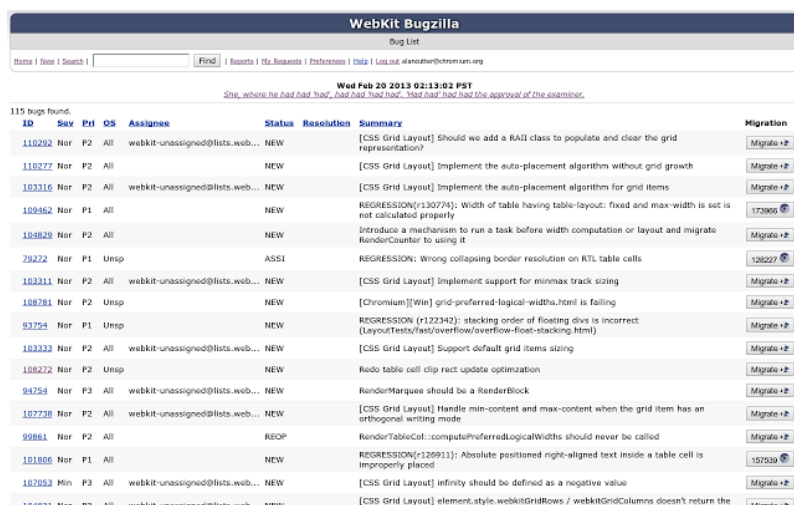
Nos dias de hoje, o número de ITS existentes é bastante acentuado, sendo que apesar de todos terem o mesmo objetivo e cobrirem na grande generalidade, as mesmas funcionalidades, existem alguns sistemas que se distinguem, como é o caso do *Bugzilla*, do *Jira* e do sistema ITS do *Github* pela sua popularidade pelo público alvo.

Bugzilla

Quando a *Mozilla* ⁹, o maior projeto *open-source* de navegadores online, precisou de um *bug tracking system*, os seus *developers* construíram o seu próprio sistema - *Bugzilla* também ele um projeto *open-source* [SC05]. No entanto, este projeto evoluiu de uma ferramenta usada internamente pelos *developers* da *Mozilla*, para um dos ITS mais utilizados pelas grandes empresas de *IT*.

⁹*Mozilla*; www.mozilla.org; 2017

Sistemas de gestão de aprendizagem e de projetos



ID	Sev	Pri	OS	Assignee	Status	Resolution	Summary	Migration
110292	Nor	P2	All	webkit-unassigned@lists.web...	NEW		[CSS Grid Layout] Should we add a RAIL class to populate and clear the grid representation?	Migrate
110277	Nor	P2	All		NEW		[CSS Grid Layout] Implement the auto-placement algorithm without grid growth	Migrate
103316	Nor	P2	All	webkit-unassigned@lists.web...	NEW		[CSS Grid Layout] Implement the auto-placement algorithm for grid items	Migrate
109462	Nor	P1	All		NEW		REGRESSION(r130774): Width of table having table-layout: fixed and max-width is set is not calculated properly	173906
104829	Nor	P2	All		NEW		Introduce a mechanism to run a task before width computation or layout and migrate RenderCounter to using it	Migrate
79272	Nor	P1	Unsp		ASSI		REGRESSION: Wrong collapsing border resolution on RTL table cells	128227
103311	Nor	P2	All	webkit-unassigned@lists.web...	NEW		[CSS Grid Layout] Implement support for minmax track sizing	Migrate
108781	Nor	P2	Unsp		NEW		[Chromium][Win] grid-preferred-logical-width.html is failing	Migrate
93754	Nor	P1	Unsp		NEW		REGRESSION(r123442): stacking order of floating divs is incorrect (LayoutTests/fast/overflow/overflow-float-stacking.html)	Migrate
103332	Nor	P2	All	webkit-unassigned@lists.web...	NEW		[CSS Grid Layout] Support default grid items sizing	Migrate
108272	Nor	P2	Unsp		NEW		Redo table cell clip rect update optimization	Migrate
94754	Nor	P3	All	webkit-unassigned@lists.web...	NEW		RenderMarquee should be a RenderBlock	Migrate
107738	Nor	P2	All	webkit-unassigned@lists.web...	NEW		[CSS Grid Layout] Handle min-content and max-content when the grid item has an orthogonal writing mode	Migrate
99861	Nor	P2	All		REQP		RenderTableCell::computePreferredLogicalWidths should never be called	Migrate
101806	Nor	P1	All		NEW		REGRESSION(r120911): Absolute positioned right-aligned text inside a table cell is improperly placed	157539
107053	Min	P3	All	webkit-unassigned@lists.web...	NEW		[CSS Grid Layout] infinity should be defined as a negative value	Migrate
104871	Nor	P3	All	webkit-unassigned@lists.web...	NEW		[CSS Grid Layout] element.style.webkitGridRows / webkitGridColumn doesn't return the	Migrate

Figura 2.1: Bugzilla Issue List

Tal como qualquer outro sistema de informação, as principais funcionalidades do *Bugzilla*, são inserir e aceder a informação. Quando um utilizador se autentica no sistema, este consegue adicionar novos *bugs*, ou pesquisar ou editar *bugs* existentes [SC05]. Um dos principais interesses dos utilizadores do *Bugzilla* prende-se com a possibilidade de verem todos os *bugs* reportados até ao momento (como podemos perceber na imagem 2.1), ver em que estado se encontram (e.g. fechado, em processamento, etc.) e perceber que resoluções foram adotadas até ao momento, se alguma tiver sido.

Jira

O ITS *Jira* tem ganho bastante popularidade nos últimos anos e oferece algumas funcionalidades únicas, como um *project management system* e a possibilidade de utilizar quadros *Kanban* no sistema [ODA⁺15].

O *Jira* é um dos ITS mais usados no dias de hoje, essencialmente porque cobre as principais funcionalidades dos sistemas ITS, funcionalidades que foram sendo referidas anteriormente, aliando também a vertente de *project management* que se torna um fator favorável à opção dos utilizadores pelo *Jira* em detrimento de sistemas concorrentes.

O *Jira* tem um quadro que é uma representação virtual dos quadros *agile kanban* bastante utilizados pelos *developers* atualmente. O *Jira* também fornece a possibilidade de manter o progresso da equipa monitorizado, gerir o *backlog* dos trabalhos e planear *sprints* [ODA⁺15].

Github

O projeto *Github*, plataforma que serve de *host* para vários projetos, foi lançada em 2008 e tem crescido desde aí, para chegar a um dos principais sites na área [BLJ⁺13]. Os dados oficiais

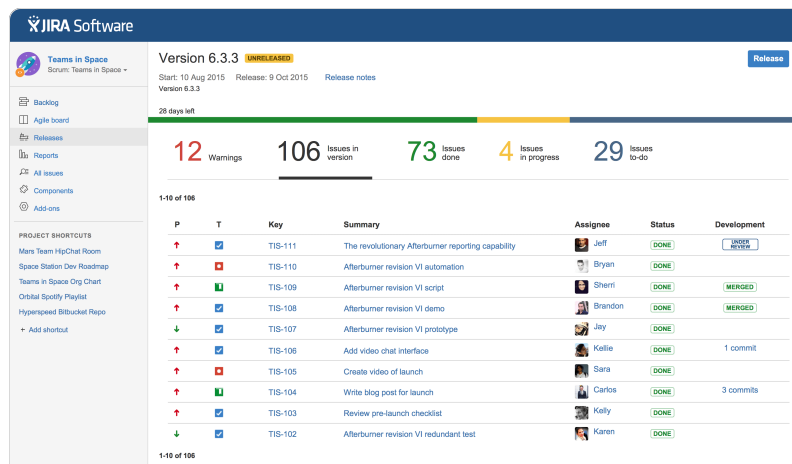


Figura 2.2: Jira project overview

da plataforma *Github* ¹⁰, mostram que este tem sido um forte competidor na área, tendo no ano de 2016 mais de 19 milhões de repositórios ativos, mais de 300 mil organizações ativas, com uma legião de utilizadores que até à data passava os 5,5 milhões *users* ativos. Para além destes números, podemos ver que o sucesso do *Github* não se mantém nos comuns *developers* em todo o mundo, mas chega também às grandes empresas internacionais, tendo sido em 2016, a *Microsoft* a empresa com mais projetos *open-source* ativos, seguindo-se mais à frente na contabilidade o *Facebook* e a *Google*.

O grande sucesso do *Github* deve-se em grande medida ao conceito em que este se insere de *social coding*, que tem criado um ambiente de *developer-friendly*, que por sua vez permite que estes se possam relacionar, colaborar e promover os seus trabalhos [BLJ⁺13]. Por exemplo, os utilizadores do *Github* podem fazer *follow* de outros membros e subscrever repositórios que queiram manter um contacto mais próximo. Estas funcionalidades distinguem o *Github* da sua concorrência e fomentam uma interação mais rica dentro da comunidade de utilizadores da plataforma e que traz um efeito positivo significativo no desenvolvimento dos projetos [AHS14].

No momento atual, existem projetos que focam mais as suas atenções no que toca às ferramentas de *Issue* ou *Bug Tracking Systems*, como no exemplo do *Jira* e do *Bugzilla*. No entanto o *Github*, pertence ao grupo de *Project hosting platforms*, que desenvolve estes sistemas de ITS, como uma ferramenta interna do seu serviço, sendo assim mais uma funcionalidade. No que toca ao número de *issues* ativos, em 2016 o *Github* teve mais de 10 milhões. O *Github* disponibiliza para cada projeto hospedado a ferramenta de ITS, que fornece as normais facilidades de um sistema deste tipo como, adicionar *issues* ou marcá-los em conformidade com a natureza de cada *issue* [BLJ⁺13]. Na imagem 2.3) é possível ver a listagem de *issues* de um repositório.

¹⁰<https://octoverse.github.com/>; 2017

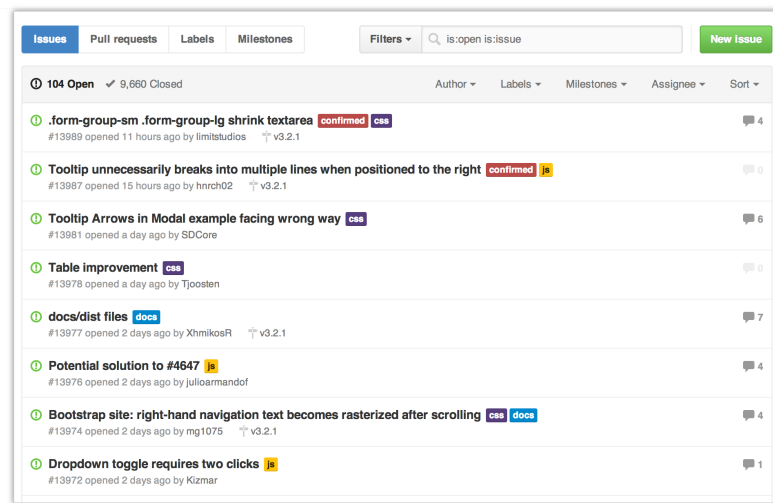


Figura 2.3: Github Issue Listing

2.3 Análise das tecnologias existentes

Hoje em dia, existem várias linguagens de programação e tecnologias de desenvolvimento que se encontram em constante crescimento. Na área do desenvolvimento *Web*, este panorama ainda parece mais correto, tendo em conta a grande quantidade de linguagens existentes que podem servir os propósitos dos programadores (e.g. *JavaScript*, *Ruby*, *Python*, *Perl*, etc.), sendo que para além das próprias linguagens, estas podem sofrer variações com a utilização das várias *frameworks* que estão constantemente a ser desenvolvidas para satisfazer diferentes critérios (e.g. para *Ruby* existe o *Ruby on Rails*, para *Python* existe o *Django*, etc.). Mesmo dentro das várias *frameworks* de uma linguagem, estas podem servir diferentes propósitos, por exemplo no caso da linguagem *JavaScript*, para programação do *server-side* podem ser usadas as *frameworks* *node.js* ou *express*, para a programação do *client-side* pode ser utilizada a *framework* *Angular* ou a *framework* *Vue.js*.

As tecnologias que foram escolhidas para proceder ao desenvolvimento deste trabalho vão de encontro à oferta que foi referida no ponto acima. O trabalho será assim desenvolvido recorrendo à *framework* *MEAN Stack*. Esta é uma ferramenta de desenvolvimento *full-stack*, o que significa que pode desenvolver uma aplicação *Web* começando na *API* e no *server-side* e terminando no *client-side*. *MEAN*, é um acrónimo que representa as quatro técnicas que esta ferramenta contempla, *Mongo.Db* (para o desenvolvimento da base de dados), *Express* (utilizado no desenvolvimento do *server-side* e da lógica do sistema), *Angular* (utilizado no desenvolvimento do *client-side*, tudo o que representa a parte visual do produto) e *Node.js* (no desenvolvimento da *API*).

Esta ferramenta tem tido um crescimento significativo no *Web Development* e isto baseia-se essencialmente no facto de as quatro *frameworks* terem como base a linguagem *JavaScript*, uma das linguagens de programação mais utilizadas na atualidade, tendo uma comunidade muito grande, que constantemente propõe melhorias às ferramentas existentes. Para além de ser uma linguagem de programação muito popular, o *JavaScript* é também uma linguagem muito dinâmica

que se enquadra em grande parte das funcionalidades que projetos como o proposto neste trabalho devem necessitar. Adicionando a isto o facto de ser uma linguagem de fácil aprendizagem e que por isso permite que este trabalho possa sofrer alterações no futuro e sempre que sejam necessárias, sem requerer um grande esforço dos seus programadores.

2.4 Conclusões

A grande vantagem de sistemas como os LMS são a existência de uma grande comunicação entre os vários utilizadores do sistema, a possibilidade de customizar o sistema, de forma a colmatar as principais necessidades para cada caso de utilização; fornecer um importante suporte à utilização das principais funcionalidades, de forma a que todos os utilizadores possam usufruir ao máximo do sistema que escolherem, sem que seja necessário dedicar muito tempo à aprendizagem da utilização de tudo isto, sem grandes custos monetários ou mesmo nenhuns.

No que toca aos sistemas ITS, estes são cada vez mais procurados pelos *developers* nos dias que correm, porque trazem uma clara mais valia para o desenvolvimento de projetos, auxiliando os mesmos a que mantenham sob controle as principais tarefas de desenvolvimento de cada projeto. Para além disto existem vários sistemas que se dedicam a fornecer este serviço como um suplemento a várias outras funcionalidades de gestão de repositórios, criando um sentido de comunidade nos seus utilizadores e fazendo com que todos se sintam parte integrante do avanço dos diversos trabalhos que estes sistemas hospedam.

Apesar destes dois tipos de sistemas terem abordagens distintas e não se relacionarem numa primeira análise, são os dois de grande importância para o desenvolvimento deste trabalho. Ao fundirmos os dois sistemas, agregando as principais funcionalidades de cada um (e.g. a facilidade de comunicação dos sistemas LMS e a possibilidade de uma comunidade identificar e resolver *issues* como nos sistemas ITS), chegamos a um pensamento que nos pode ajudar a desenvolver um sistema, que sirva não só para unir a comunidade da instituição em questão, mas também que consiga de forma rápida e eficaz resolver os problemas que nesta instituição vão sendo levantados ao longo do tempo.

Capítulo 3

Sistema de Informação para o acompanhamento curricular

3.1 Modelo atual utilizado na FEUP

Na Faculdade de Engenharia da Universidade do Porto (FEUP), cada vez se torna mais importante preservar os níveis de qualidade internos e ter em atenção os assuntos que dizem respeito tanto ao corpo docente da Faculdade, como também aos alunos.

E por isso mesmo, é hoje uma tarefa com muito relevo, proceder à auscultação de todos os intervenientes da FEUP, para tentar perceber quais são os pontos que devem ser mantidos e aqueles que devem ser melhorados no normal funcionamento das atividades da Faculdade.

Para além de ser dada muita importância à opinião de todos os intervenientes, também é necessário manter sob controlo todas as situações que possam originar algum problema no dia a dia da faculdade e intervir sempre que necessário, de forma a não deixar estas situações de risco alastrar-se.

A atenção que é exigida aos vários intervenientes que possam ter responsabilidades na manutenção destes critérios (secretariado, docentes, diretores de curso e departamento, órgãos de gestão da FEUP, etc.) é cada vez maior.

No entanto, na FEUP existe um órgão de gestão que é responsável por exercer o primeiro contacto com os vários grupos de atores ativos na comunidade, de forma a tentar preservar todos estes critérios de satisfação que a FEUP define para si própria, que são as comissões de acompanhamento.

Estas comissões de acompanhamento são órgãos eleitos, compostos por docentes e alunos (eleitos respetivamente por docentes e alunos), que têm a seu encargo manter o bom funcionamento das atividades do departamento pelo qual foram eleitos. É pedido aos elementos que constituem as comissões de acompanhamento, que mantenham um contacto direto entre a comunidade que constitui o departamento que representam, que engloba essencialmente os docentes, os alunos e o secretariado.

Em cada departamento, estas comissões podem ter responsabilidades e pesos diferentes. Algumas destas responsabilidades vão desde intermediar o contacto entre os diferentes grupos de atores do departamento (docentes, alunos e secretariado), passando por identificar problemas e sugerir formas de resolução para os mesmos, até situações mais logísticas do próprio departamento (marcação de exames, organização de atividades extra-curriculares no âmbito do curso, etc.).

Cada vez mais, é dada uma importância extra a este tipo de tarefas e por conseguinte, a estas comissões de acompanhamento, que auxiliam em grande medida a que o bom funcionamento da Faculdade seja preservado e que possam ser implementadas melhorias em várias atividades do dia a dia da FEUP.

3.2 Problemas do modelo atual

Apesar do sistema que é mantido atualmente na FEUP ser de grande importância para o bom funcionamento da comunidade, existem alguns problemas que podiam ser melhorados e que este trabalho se propõe a resolver.

Um dos maiores promotores da manutenção dos níveis de qualidade a que a FEUP se propõe cumprir, é a comunicação que os vários intervenientes da comunidade devem manter entre si. No entanto, o facto de os vários intervenientes poderem ter uma maior facilidade de contacto de formas diferentes (contacto pessoal, *email*, *skype*, etc.) dificulta este contacto.

Para além disto, é importante poder assegurar que os problemas que são reportados (quer seja ao secretariado, a algum docente em particular ou a membros da comissão de acompanhamento), sejam encaminhados para as pessoas responsáveis ou resolvidos se for caso disso, de forma atempada e para evitar que alguns problemas se propaguem e se tornem de mais difícil resolução. Resolver os problemas que vão sendo reportados de forma atempada, também permite que mais interessados possam dar a sua sugestão e que a solução adotada seja do interesse de todos ou de uma grande maioria.

Várias vezes os problemas que são reportados, são idênticos a problemas que foram resolvidos no passado. Porém na maior parte das vezes, estes problemas são reportados pessoalmente, e torna-se difícil manter um histórico das situações que vão ocorrendo ao longo do tempo e das respetivas soluções que vão sendo adotadas. No entanto, manter um histórico destas situações seria uma mais valia, para poder tomar decisões consistentes e também para poder perceber se uma solução para um problema recorrente teve efeitos práticos no passado ou não.

Uma das componentes principais do acompanhamento curricular, é possibilitar aos vários membros da comunidade da FEUP, que possam participar na manutenção de qualidade da faculdade, sugerindo melhorias para o normal funcionamento das atividades. No entanto, se estas mudanças não puderem ser aplicadas no momento atual, podem ficar esquecidas e não chegar a ser aplicadas quando necessárias.

3.3 Integração de um sistema de informação para o acompanhamento curricular

De forma a colmatar alguns dos problemas existentes no modelo que atualmente é aplicado na FEUP, alguns dos quais foram sendo referidos nos pontos anteriores, será importante que o sistema que este trabalho pretende elaborar, perceba quais são as principais funcionalidades a implementar, quais são os intervenientes a que o sistema pretende chegar, quais são as mais valias a ter em atenção e quais são os principais métodos de validação do sistema elaborado.

3.3.1 Principais Funcionalidades

Sistema aberto à comunidade

O sistema pretende ser aberto à comunidade da FEUP. Podendo ser utilizado por qualquer aluno, docente ou membro do secretariado que pretende dar conta de um problema ou situação de risco que tenha identificado, ou até de forma a poder dar a sua opinião ou deixar uma sugestão sobre algum tema que lhe possa dizer respeito ou que o possa afetar de alguma forma no dia a dia. Desta forma o sistema pode equiparar-se ao modelo que atualmente é posto em prática na FEUP, onde qualquer elemento da comunidade FEUP, pode fazer uma sugestão ou dar conta de um problema a um membro da comissão de acompanhamento, a um docente ou a um membro do secretariado por exemplo.

Por isto mesmo, é importante que os *issues* levantados (quer seja uma sugestão ou um problema), sejam agrupados por temas específicos, que sejam de fácil consulta e que sejam encaminhados para os responsáveis que poderão dar a devida atenção ao *issue* e resolvê-lo se for caso disso. No entanto, é de grande importância que estes *issues* apenas possam ser resolvidos pelas pessoas responsáveis em cada grupo.

Grupos

Sendo assim, importa definir mais concretamente o conceito de grupos. Estes grupos servem para dividir os membros da comunidade da FEUP e agregá-los da forma mais conveniente para todos. Esta divisão deverá ser feita primeiramente entre os vários cursos da FEUP. Depois podem ser feitas divisões por unidades curriculares (UC), contendo os alunos inscritos a essa UC, mais os docentes responsáveis pela mesma. Também deverão existir grupos que agreguem os alunos de um certo ano curricular, facilitando assim o diálogo entre os alunos que frequentem em grande parte as mesmas UC.

A existência destes grupos, permite que os membros da comunidade da FEUP possam ter fácil acesso a informações que lhes digam respeito ou que sejam próximas e impossibilita os mesmos de poderem ver situações que ocorram dentro de grupos onde não estejam inseridos.

No entanto, será necessário inserir nos vários grupos que vão sendo criados, alguns dos membros que tenham responsabilidade no que toca à comunicação e resolução de *issues*. Estes elementos podem ser alunos representantes de ano, membros das comissões de acompanhamento, os diretores de curso e departamento e membros do secretariado do curso em questão.

Histórico

Um dos principais benefícios que este tipo de sistema pode trazer, é o facto de todos os *issues* levantados, poderem ser guardados num histórico. Este registo será acessível em cada grupo criado e deverá conter a informação mais pertinente de cada *issue*, membro que o reportou, solução que foi tomada, por quem foi solucionado e outros comentários que forem sendo adicionados durante o seu processo de resolução.

Num nível mais alto de gestão do sistema, é possível ver um histórico mais geral dos *issues* reportados o que pode auxiliar a que soluções idênticas possam ser tomadas para problemas recorrentes em grupos diferentes.

Tags

Para ser possível organizar melhor os *issues* do sistema, é importante definir um sistema de *labeling* que possa ser facilmente associado a um tema, grupo ou estado. Um *issue* pode ter várias etiquetas (*tags*) associadas, desde que estas façam sentido na caracterização do *issue* em questão e desde que este *labeling* não torne o processo de organização dos vários *issues*, um processo menos intuitivo e prático. Sendo assim o sistema deve disponibilizar um conjunto de *tags* pré-definidas para facilitar o processo de *labeling* dos *issues* que forem sendo levantados no sistema, mas também é importante permitir que um novo conjunto de *tags* sejam adicionados posteriormente, se estas fizerem sentido.

Uma das principais utilizações destas *tags* pode ser para agrupar um conjunto de *issues* por tema (e.g. exames, horários,...), ou para dar a indicação do estado atual do *issue* (e.g. por analisar, em análise, concluído).

Página de um *issue*

Para ser mais fácil resolver os *issues* que vão sendo identificados, estes devem ficar numa lista de *issues* de um certo grupo e cada um deve ter uma página própria, onde seja possível ver informações úteis que vão sendo disponibilizadas enquanto o *issue* não for fechado.

Nesta página deve ser possível ver o estado do *issue*, ver quem são os utilizadores com permissões para o ver e para o resolver respetivamente, a data de início e quem o reportou e o grupo em que está inserido. Para além disso deve existir uma secção de comentários, para poder existir uma troca de ideias durante o processo de resolução.

Quando o *issue* estiver concluído, deve ser adicionado o método de resolução, de forma a poder ser guardado no histórico de *issues* e para ser mais facilmente consultado posteriormente. No estado de concluído, não deve ser possível adicionar mais informações nem comentários.

Dashboard pessoal

Para possibilitar uma maior facilidade de acesso ao sistema para todos os utilizadores, devem ser disponibilizados *dashboards* personalizados para todos, onde é apresentada a principal atividade de todos os grupos onde os utilizadores estão inseridos.

No caso de um utilizador ser um dos responsáveis por assegurar a gestão de algum grupo, no seu *dashboard* serão notificadas todas as atividades que digam respeito a esses grupos (e.g. um docente encarregue por uma UC a que tenha sido reportado um *issue*, um aluno representante de ano no grupo do seu ano curricular). Estas notificações servem para manter os responsáveis a par de qualquer atividade que ocorra nos vários grupos que possam ter associados.

Para um membro não gestor de nenhum grupo, este *dashboard* serve para receber notificações de atividade pública que tenha ocorrido num grupo em que esteja inserido (e.g. alteração da data de um teste a uma UC em que esteja inscrito).

3.3.2 Mais valias do sistema

Apesar de todas estas funcionalidades completarem um conjunto de atividades, que um sistema que se propõe a fazer o acompanhamento curricular de uma instituição com tantos elementos como a FEUP deve ter, existem algumas funcionalidades que são vistas como uma mais valia, no que toca à resolução de alguns problemas que existem no modelo que é utilizado hoje em dia.

Uma das principais mudanças que este sistema pode trazer ao modelo de acompanhamento curricular, é a possibilidade de analisar casos idênticos aos que vão sendo levantados, tentando adotar uma solução semelhante às que foram sendo adotadas em casos passados. Para isto o histórico tem uma grande importância, pois pode facilitar e agilizar o processo de resolução de problemas frequentes que vão sendo reportados ao longo do tempo.

Um dos principais focos deste trabalho é fazer com que todos os membros da comunidade da FEUP possam manter o contacto constante e se sintam impelidos a utilizar o sistema sempre que achem que se justifique. Isto é uma das grandes vantagens do modelo atual, em que qualquer aluno ou docente pode entrar em contacto com o membro responsável por um determinado assunto e fazer uma sugestão de melhoria ou reportar um problema que tenha identificado. No entanto, o sistema que propomos implementar, pode facilitar em muito este contacto. Uma consequência deste sistema estar disponível a todos e permitir um levantamento dos *issues* instantaneamente, faz com que estes possam ser resolvidos de forma mais rápida e eficaz, sem permitir que os problemas se propaguem ou se tornem mais complicados de resolver no futuro.

O facto de cada utilizador ter uma página pessoal, onde possa ver toda a atividade que tem ocorrido nos grupos onde está inserido, é uma grande vantagem para dar conhecimento de forma rápida a todos os utilizadores do sistema. Para além disto, os utilizadores com responsabilidades em algum grupo onde estejam inseridos, têm a possibilidade de receber notificações de situações que possam requerer a sua atenção, permitindo que os *issues* possam ser resolvidos mais rapidamente.

Este método também permite que o sistema seja mais intuitivo e fácil de utilizar, sem que o utilizador necessite de ir a todos os grupos em que seja um membro, para ter acesso às últimas atividades.

3.3.3 Métodos de validação do sistema

De forma a garantir que este sistema suprima as necessidades dos utilizadores, é importante estruturar desde cedo um grupo de testes ao sistema que permita perceber quais são as principais necessidades dos futuros utilizadores do sistema. Sendo assim, é importante que numa fase inicial, tanto os membros da comissão de acompanhamento, como alguns membros do secretariado possam testar as principais funcionalidades já implementadas e que constituem o mínimo produto viável a desenvolver. Apesar de ser um dos objetivos apontados ao sistema, poder ser uma ferramenta que auxilie o funcionamento em toda a comunidade da FEUP, de forma a poder manter um ambiente de testes o mais controlado possível, é importante focar as atenções numa fase inicial para o Departamento de Engenharia Informática (DEI). Sendo assim, podemos acompanhar mais de perto o desenrolar dos testes e tentar colmatar as necessidades e fazer as alterações que vão sendo necessárias ao longo do tempo. Indo de encontro ao que foi estabelecido no ponto acima mencionado, também o número de alunos e docentes será reduzido numa fase inicial, de forma a podermos testar o sistema para algumas UC, tendo em conta que depois de obtermos o funcionamento esperado que um grupo deve ter, depois resta replicar o processo para todas as UC do sistema. Numa fase mais avançada do desenvolvimento, é necessário fazer testes mais abrangentes, já com casos reais de utilização, potencialmente com mais alunos e docentes. Focando ainda as atenções no DEI, mas já incorporando várias UC, de forma a testar grande parte, senão mesmo, a totalidade das funcionalidades do sistema.

3.3.4 Melhoramentos ao sistema proposto

Este sistema pretende essencialmente criar uma forma de melhorar o acompanhamento curricular utilizando como base algumas funcionalidades do modelo utilizado atualmente na FEUP. No entanto, a intenção deste projeto é definir um caminho que possa ser seguido na elaboração de ferramentas que possam ser desenvolvidas num futuro próximo para apoiar o dia a dia dos membros da comunidade da FEUP. Isto passa por integrar mais os alunos, docentes e secretariado no sistema que é proposto nesta dissertação, adicionando funcionalidades que lhes sejam úteis no seu trabalho quotidiano. Uma das propostas de futuro é a possibilidade de adaptar o sistema para que existam mensagens instantâneas, diretas e de grupo, de forma a replicar a importância que, por exemplo as redes sociais têm nos dias de hoje nos vários grupos de alunos. Sendo assim, seria possível que os vários intervenientes pudessem partilhar sugestões ou conhecimento de forma mais rápida e prática utilizando um sistema interno da faculdade. Outro melhoramento possível seria a integração com ferramentas de partilha de ficheiros na cloud (eg. *Google Drive*, *Dropbox*, etc.), possibilitando que os conteúdos letivos que os alunos e professores vão desenvolvendo (e.g. apontamentos, resoluções de exercícios das aulas, etc.) pudesse ser partilhado de forma rápida e

Sistema de Informação para o acompanhamento curricular

segura. Uma forma de tornar o sistema mais apelativo para aqueles que o utilizam, é tornar as páginas pessoais de cada um, ainda mais personalizadas. Adicionando por exemplo a informação dos horários de cada um, ou um calendário de exames e pontos de avaliação.

Capítulo 4

Solução Desenvolvida

Depois de proceder à análise extensiva do sistema que é utilizado atualmente na FEUP para proceder ao levantamento e resolução de problemas, é-nos possível dar início à fase de implementação e desenvolvimento de um sistema que seja capaz de suprir as necessidades que foram sendo identificadas anteriormente no capítulo 3. A plataforma desenvolvida acolhe as principais funcionalidades que um sistema com este objetivo deve ter e acrescenta algumas novas funcionalidades que auxiliam a rápida observação e resolução de problemas.

Este capítulo pretende dar a conhecer a arquitetura implementada no sistema, descreve como está construída a plataforma, particularizando toda a estrutura da implementação. Para além disto, são descritas e explicadas todas as funcionalidades desenvolvidas e de que forma devem apoiar o sistema. Também será apresentado o fluxo de utilização do sistema construído .

4.1 Arquitetura do sistema

O sistema desenvolvido segue uma arquitetura de *client-server* comum, onde o servidor é responsável por receber e tratar de todos os pedidos e lógica executada pelos clientes da plataforma. O servidor é responsável por receber os pedidos dos utilizadores da plataforma, tratar da lógica do sistema interpretando os pedidos e retornando a informação já processada, e também gerir a informação existente na base de dados da plataforma. Estes dados estão guardados numa base de dados externa, tendo em si inseridos estruturas de dados essenciais à manutenção e utilização esperada da plataforma. Dados estes que serão descritos posteriormente neste capítulo, na secção 4.3.3.

A plataforma construída não utiliza serviços externos, à parte do serviço de base de dados, pois não tem qualquer interoperabilidade com o sistema de informação da Faculdade (*Sigarra*), desta forma os dados existentes na plataforma são independentes de outros sistemas utilizados.

O diagrama representado na figura 4.1 descreve a arquitetura do sistema desenvolvido.

Solução Desenvolvida

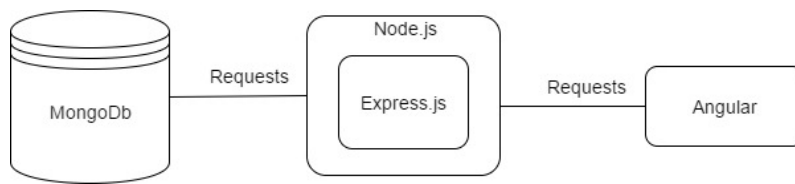


Figura 4.1: Arquitetura do sistema

4.2 Tecnologias

A plataforma foi construída fazendo uso de várias tecnologias, que foram escolhidas com o principal objetivo de produzir resultados práticos e para que a efetiva utilização da plataforma fosse possível no final da implementação. Desta forma, as principais tecnologias utilizadas são:

- MEAN Stack: MEAN é uma *framework full stack* baseada em *JavaScript*, que tem como objetivo simplificar e acelerar o desenvolvimento de plataformas *web* [GSHH16]. O nome MEAN vem da conjugação das quatro ferramentas, que em conjunto oferecem uma solução, tanto para componentes de aplicações *web client*, como para *server-side*, *MongoDb*, *Express.js*, *Angular* e *Node.js*. Todas estas ferramentas são baseadas na linguagem *JavaScript*, que apesar de ter sido inicialmente desenvolvida para soluções *client-side*, começou a ser comumente utilizada também para soluções de *server-side*, devido em grande parte ao crescimento do *Node.js* [PJC15].
 - MongoDB: É uma base de dados *NoSQL* de alta performance, construída fazendo uso do formato de dados *JSON* e é, por isso, ideal para sistemas *server-side* construídos utilizando *JavaScript* como principal linguagem [PJC15]. MongoDB é *document-oriented*, não sendo assim uma base de dados relacional, sendo o principal objetivo desta característica, permitir que a base de dados seja mais facilmente escalável [Cho13].
 - Express: É uma *framework* minimalista e flexível baseada em *Node.js*, que permite uma série de funcionalidades robustas de desenvolvimento de aplicações *web* e *mobile* [exp].
 - Angular: É uma plataforma para construção de aplicações *web* e *mobile*, que usa *TypeScript* como linguagem fonte da *framework* [anga]. Angular é uma *framework open-source*, mantida pela Google, que permite a criação de aplicações *client-side* usando o modelo MVC [PJC15].
 - Node.js: É um ambiente de execução construído em *JavaScript*, que permite a construção de *network applications*, facilmente escaláveis [nod]. Mesmo havendo várias soluções que permitem a construção de uma *API REST*, ao usar *Node.js & Express* aumentamos a produtividade, diminuindo o esforço de desenvolvimento, enquanto é desenvolvido um sistema eficaz, eficiente e altamente escalável [PJC15].

- mLab: É uma base de dados completamente gerida por um serviço *cloud*, que auxilia ao provisionamento automático, escalabilidade, *backup*, monitorização, ferramentas de gestão e suporte de bases de dados construídas em *MongoDb* [mla]
- Bootstrap: é uma *framework* popular de *HTML*, *CSS* e *JavaScript*, para desenvolvimento de aplicações responsivas, moldadas à utilização móvel para *web* [boo]. Em conjunto com *Angular*, é possível criar uma *web application* poderosa facilmente e enriquecida pela capacidade de design que a utilização destas *frameworks* traz [PJC15].
- Angular-cli: é uma *Command Line Interface* (CLI) que automatiza o *workflow* de desenvolvimento [angb]. Utilizando esta interface é simplificada a criação de aplicações em *Angular* e várias outras funcionalidades existentes em aplicações construídas em *Angular*.
- JavaScript: é uma linguagem utilizada para desenvolvimento *web* e a grande maioria dos *websites* atuais utilizam esta linguagem como linguagem fonte ou como principais interpretadores de código, fazendo desta linguagem a mais ubíqua na história da programação [Fla06].
- TypeScript: é uma linguagem com base em *JavaScript* para aplicações com necessidades de escalabilidade. Esta linguagem acrescenta tipos, classes e módulos adicionais às já existentes em *JavaScript*. *TypeScript* compila para *JavaScript* legível e baseado nos seus *standards* [typ].

É feito o uso de todas estas tecnologias na produção da plataforma proposta nesta dissertação e na secção seguinte será explicado de forma mais concreta, em que contextos foram utilizadas.

4.3 Plataforma

4.3.1 Servidor

Tal como já tinha sido descrito anteriormente, o servidor permite que a plataforma seja constantemente alimentada com informação pertinente, em cada momento de utilização. Esta componente do sistema construído, é responsável por tratar todos os pedidos vindos da camada do cliente, interpretar esses pedidos e se necessário pedir informação necessária ao correto funcionamento da aplicação, à base de dados.

Para funcionar de forma eficiente e eficaz, este servidor foi construído seguindo a estrutura *REST API*, o que permite que os vários pedidos sejam tratados de forma correta e segura, e que a informação possa ser corretamente obtida e atualizada na base de dados se assim for necessário.

Representational State Transfer (*REST*) utiliza as operações nativas *HTTP* - *POST*, *GET*, *PUT* e *DELETE* - que se ligam diretamente às quatro funções principais da utilização comum de uma base de dados: Criar, ler, editar e remover. Estas funcionalidades, podem assim, ser utilizadas depois numa utilização comum do serviço que for criado [PJC15].

Sendo assim, a mais valia do servidor construído para este sistema, prende-se com o facto de o utilizador ter acesso a informação constantemente atualizada - garantindo apenas que exista uma ligação à *internet* - quer seja sobre os *issues* abertos no momento, ou informações mais sensíveis, que possam ser restritas a utilizadores com níveis de permissões mais elevadas.

4.3.2 Cliente

Um dos principais requisitos da camada de cliente desta plataforma, é a capacidade de ter um sistema dinâmico que se adapta à informação que vai recebendo do utilizador do sistema e altera a informação que é mostrada. Sendo assim a utilização de *Angular* como *framework* de *JavaScript*, permite que o dinamismo exigido a uma plataforma deste tipo, seja alcançado e que o sistema seja intuitivo e fácil de utilizar pelos seus utilizadores.

Na plataforma existem três grandes secções que são divididas na implementação, que são o *dashboard* da aplicação - que mostra ao utilizador os seus grupos e dependendo dos níveis de privilégios do mesmo, pode ou não ter funcionalidades de administrador ou gestor do sistema -, a página de um grupo - que contém uma listagem de todos os *issues* que pertencem a este grupo - e a página de um *issue* - que contém as informações principais deste *issue* e todos os comentários adicionados ao longo do tempo neste *issue*. Estas páginas têm atividades diferentes, consoante os níveis de permissões do utilizador que as está a ver e este é por isso, um bom exemplo do dinamismo que pode ser dado a uma plataforma como esta.

4.3.3 Estrutura de dados

Apesar de o sistema fazer uso de um grande número de dados, a estrutura de dados criada para suportar esta plataforma, não comporta uma grande complexidade. Isto porque os dados que são necessários armazenar, apesar de poderem ser grandemente replicados, são de composição direta e intuitiva.

Os tipos de dados guardados pelo sistema são de quatro tipos, *users*, *issues*, *comments* e *groups*. Para os dados referentes aos utilizadores do sistema, existe a tabela *users*, que contém o *username* do utilizador, a *password* encriptada. Existe também o atributo *role*, que se refere ao papel que este utilizador tem no sistema, que pode ser do tipo *student* - o nível com menos permissões do sistema -, *manager* - um utilizador com permissões de gestão para grupos específicos do sistema - e *admin* - que é a posição mais alta de gestão. Existem ainda mais dois atributos para a classe de user, *manager* e *permissions*, que são respetivamente, uma lista de todos os grupos que este utilizador gere e uma lista de todos os grupos a que este utilizador tem permissões de visualização (estas listas podem estar vazias, independentemente do papel do utilizador no sistema).

```
{  
  "username": "Jorge Teixeira",  
  "password": "$2a$10$Gg1HNn7pWR4pG1pmA3heue5seGdI1.aNotliHd3XXXXXXXXXXXXX",  
  "role": "manager",  
  "manager": [  

```

```

        "EIC0083",
        "EIC0004",
        "EIC0028"
    ],
    "permissions": [
        "EIC0083"
    ]
}

```

Os dados correspondentes aos *issues* criados pelo sistema, são também um dos tipos de dados guardados pela base de dados. Esta tabela guarda o título do *issue* (uma pequena descrição do que é pretendido expôr com a criação deste *issue*), o nome do utilizador que criou o *issue*, o estado em que o *issue* se encontra (pode ser aberto, em tratamento ou fechado), o tipo de *issue* (pode ser um problema, uma proposta, ou uma nota), o *id* do grupo a que este *issue* pertence, uma lista de *tags* a que o utilizador decidiu atribuir este *issue* e a data de criação do *issue*.

```

{
    "title": "Também posso abrir e editar um issue",
    "creator": "Jorge Teixeira",
    "state": "Solved",
    "type": "Note",
    "uc": "EIC0000",
    "tags": [],
    "creation_date": "2017-06-10T19:24:36.055Z"
}

```

A tabela correspondente a um grupo, utiliza para seu benefício a possibilidade das estruturas de dados em *MongoDb* poderem ser dinâmicas, dependendo do objeto que quisermos adicionar à base de dados em cada momento. No caso de um grupo, isto torna-se importante, porque um elemento desta tabela tanto pode ser do tipo UC (unidade curricular), ou pode ser um grupo externo que o administrador tenha criado, num determinado momento (*e.g.* grupo para questões de horários). Sendo assim, os atributos comuns de um elemento desta tabela são: o *id* do grupo, que no caso de uma UC pode ser o *id* que esta tem no sistema de informação da FEUP, noutra caso pode ser um *id* que faça sentido em cada grupo; o nome do grupo criado. Os atributos que existem em grupos que são interpretados como UC, contêm também o ano e o semestre da UC a que pertencem. Para além destes atributos, existe um último atributo que é um booleano que se refere a um grupo e retorna verdadeiro se este grupo for uma UC e falso se não for.

```

{
    "uc": true,
    "id": "EIC0000",
    "name": "Unidade Curricular do 1º Ano",

```

```
"sem": 1,
"year": 1
}
```

A tabela correspondente aos comentários existentes na página de um *issue*, é constituída pelos seguintes atributos: o *id* desse comentário, que é um valor incrementado automaticamente sempre que um comentário é adicionado; o corpo do comentário; o nome do autor que fez o comentário; o *id* do *issue* a que o comentário pertence; a data de criação do comentário.

```
{
  "id": 1,
  "body": "Fica aqui um exemplo de comentário",
  "author": "Jorge Teixeira",
  "issueID": "593c4774afad314768554b37",
  "date": "2017-06-10T19:24:57.138Z"
}
```

De forma a que toda a informação importante do sistema esteja constantemente assegurada e acessível para os utilizadores da plataforma, foi criada uma base de dados num serviço externo, o *mLab*.

4.3.4 API

O servidor, explicado na secção 4.3.1 de forma mais extensa, disponibiliza uma *REST API*, que permite que o sistema construído, possa aceder a toda a informação que a plataforma deve conter para o seu uso comum. A vantagem da criação de uma *API* deste género, é a possibilidade de o serviço cliente do projeto, poder sofrer alterações profundas, sem necessitar de uma atualização que diga respeito aos dados criados e mencionados na secção anterior. Desta forma, é tanto possível criar uma plataforma *web* que cumpra os requisitos iniciais, como uma plataforma móvel ou híbrida, utilizando a mesma base em todos os casos.

De forma a garantir que estes dados não possam ser visualizados por utilizadores sem as permissões respetivas, todos os pedidos à *API* devem ser feitos sempre contendo um *token* de identificação, gerado pelo servidor, que garante ao sistema que o utilizador que solicita um pedido, tem autorização para o fazer. Para que a plataforma seja completamente preenchida com todos os dados necessários, é preciso proceder a vários pedidos à *API* e estes serão descritos com uma maior profundidade de seguida.

Pedidos relacionados com *Issues*:

- **GET/issues:** Recolhe todos os *issues* inseridos na base de dados pertencentes a todos os grupos;
- **GET/issues:id:** Recolhe o *issue* desejado, inserindo apenas o seu *id*. Retorna uma mensagem de erro, caso não exista o *issue* com o *id* em questão;

Solução Desenvolvida

- **GET/issues/uc:id:** Recolhe todos os *issues* existentes na base de dados, que estejam associados a um determinado grupo, bastando inserir o *id* desse grupo;
- **POST/issues:** Adiciona um novo *issue* à tabela de *issues* existentes na base de dados. Basta inserir o objeto correspondente ao *issue* a inserir no pedido;
- **PUT/issues:id:** Atualiza um determinado *issue* na base de dados, tendo que inserir o *id* de identificação do *issue* a editar e passando um objeto com as novas informações do *issue*;
- **DELETE/issues:id:** Elimina o *issue* da base de dados com o *id* passado no pedido.

Pedidos relacionados com *Users*:

- **GET/users:** Recolhe todos os *users* inseridos na base de dados;
- **GET/users:id:** Recolhe o *user* desejado, inserindo apenas o seu *id*. Retorna uma mensagem de erro, caso não exista o user com o *id* em questão;
- **POST/users:** Confirma a existência de um *user* com os parâmetros inseridos por um utilizador na secção de *login*. Este pedido retorna uma mensagem de sucesso ou de erro, consoante o resultado obtido na base de dados;
- **POST/users/new:** Adiciona um novo *user* à tabela de users existentes na base de dados, quando um novo utilizador efetua um registo. Basta inserir o objeto correspondente ao *issue* a inserir no pedido;
- **PUT/users:id:** Atualiza as informações de um determinado user na base de dados, tendo que inserir o *id* de identificação do user a editar e passando um objeto com as novas informações do *user*;
- **DELETE/users:id:** Elimina o user da base de dados com o *id* passado no pedido.

Pedidos relacionados com *Groups*:

- **GET/uc:** Recolhe todos os grupos inseridos na base de dados;
- **GET/uc:id:** Recolhe o grupo desejado, inserindo apenas o seu *id*. Retorna uma mensagem de erro, caso não exista o grupo com o *id* em questão;
- **POST/uc:** Adiciona um novo grupo à tabela de ucs existentes na base de dados. Basta inserir o objeto correspondente ao grupo a inserir no pedido.

Pedidos relacionados com *Comments*:

- **GET/comment:id**: Recolhe o *comment* desejado, inserindo apenas o seu *id*. Retorna uma mensagem de erro, caso não exista o comment com o *id* em questão;
- **GET/comment/count:id**: Retorna o número de *comments* que um grupo contém, passando o *id* do grupo desejado;
- **POST/comment**: Recolhe todos os *comments* inseridos na base de dados;
- **PUT/comment:id**: Atualiza as informações de um determinado *comment* na base de dados, tendo que inserir o *id* de identificação do *comment* a editar e passando um objeto com as novas informações do *comment*;
- **DELETE/comment:id**: Elimina o *comment* da base de dados com o *id* passado no pedido.

4.4 Funcionalidades implementadas

A intenção principal do desenvolvimento deste sistema é fazer com que as funcionalidades implementadas possam suprir as necessidades dos seus utilizadores, assim que esta esteja finalizada. Portanto, as funcionalidades que foram desenvolvidas, são uma representação bastante aproximada daquilo que a plataforma deveria ter numa situação ideal.

O sistema está preparado para aceitar utilizadores com diferentes níveis de permissões, sendo assim, as funcionalidades que serão descritas de seguida vão ser divididas por níveis de permissões. As permissões, por ordem crescente de importância, são: *Student*, *Manager* e *Admin*; e pressupõe-se que um utilizador com níveis de permissões mais avançadas, tem acesso às mesmas funcionalidades que um utilizador com menos permissões. Desta forma as funcionalidades implementadas são:

Funcionalidades de *Student*:

- O utilizador consegue ver no seu *dashboard*, todos os grupos a que este pertence - *e.g* Um estudante inscrito a uma unidade curricular, pode vê-la no seu *dashboard*;
- O utilizador pode ver os *issues* de um grupo a que pertence, na página deste grupo. Pode ver que utilizador os criou e em que data foi criado;
- O utilizador pode adicionar *issues* a um grupo a que pertence;
- O utilizador pode editar *issues* que adicionou, alterando a descrição do mesmo e o tipo de *issue* (Problema, proposta ou nota);

Funcionalidades de *Manager*:

- O utilizador pode ver os grupos que gere - *e.g*. Um gestor de uma unidade curricular, pode ver numa secção idêntica à dos seus grupos, uma lista dos grupos que gere;

- O utilizador pode ver os *issues* dos grupos que gere;
- O utilizador pode ver os comentários dos *issues* pertencentes a grupos que gere;
- O utilizador pode adicionar comentários a *issues* que gere;

Funcionalidades de *Admin*:

- O utilizador tem acesso a um *dashboard* mais geral, com todos os grupos existentes no sistema;
- O utilizador tem acesso a um semáforo que monitoriza os grupos existentes no sistema e caracteriza-os como verde, amarelo ou vermelho, dependendo dos *issues* abertos;
- O utilizador pode ver estatísticas do sistema, tal como um diagrama que mostra em que meses foram abertos mais *issues*, qual é o grupo com mais *issues* abertos, entre outros;
- O utilizador pode editar dados da gestão dos semáforos que existem no seu *dashboard* geral;
- O utilizador pode adicionar grupos novos ao sistema;
- O utilizador pode fechar um *issue*, considerando-o resolvido;
- O utilizador pode editar um *issue* já aberto, mesmo que não seja ele o criador;
- O utilizador pode editar o papel dos utilizadores do sistema, alterando-lhes as suas permissões;
- O utilizador pode adicionar um utilizador como gestor de um grupo;
- O utilizador pode adicionar permissões de visualização de um grupo a um utilizador;
- O utilizador pode eliminar utilizadores do sistema;
- O utilizador pode eliminar *issues* de um grupo.

Para além destas funcionalidades, o sistema permite o registo de novos utilizadores, sendo necessário que um administrador do sistema valide este utilizador posteriormente. Neste momento de validação, o administrador deve atribuir um papel ao utilizador recém registado.

4.5 Fluxo de utilização

Uma necessidade expressa de um sistema que se predisponha a resolver problemas de forma rápida e eficaz, é a possibilidade de reportar e resolver estes mesmos problemas de forma rápida e intuitiva no sistema. Sendo assim, um utilizador que pretenda reportar um problema, um gestor que pretenda ver qual é a situação atual dos seus grupos e um administrador que pretenda fazer a

manutenção do sistema, não precisará de passar por muitas páginas e proceder a muitas ações para poder ver resultados.

Desta forma, as principais ações do sistema são:

1. Inicialmente, alguém que aceda ao sistema e não esteja autenticado, poderá ver a página inicial - figura 4.2 - onde existem apenas as opções de fazer *login* e registrar no sistema, caso ainda não o tenha feito. Para proceder ao *login* no sistema, basta inserir o *username* e *password* correspondente aos seus dados de utilizador.

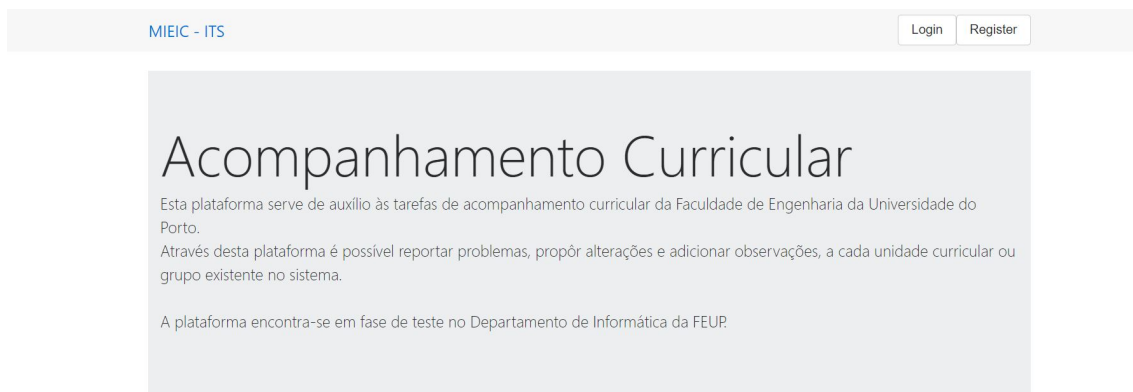


Figura 4.2: Página inicial

2. Depois de proceder ao *login* ou ao registo, o utilizador terá acesso ao seu *dashboard* pessoal. Neste passo existem quatro possibilidades:

- (a) Um utilizador acabado de se registar não está ainda validado, sendo assim a página do seu *dashboard* pessoal, dá-lhe conta desta mesma situação com um aviso - como podemos ver na figura 4.3.

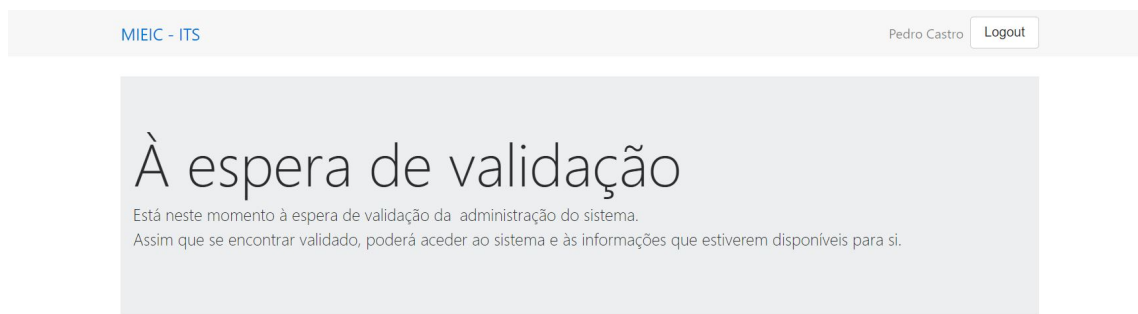


Figura 4.3: Utilizador não validado

- (b) Um utilizador já validado, que seja *Student*, poderá ver no seu *dashboard* pessoal a listagem de todos os grupos a que pertence e dos quais tem permissões de utilizador - situação descrita na figura 4.4.

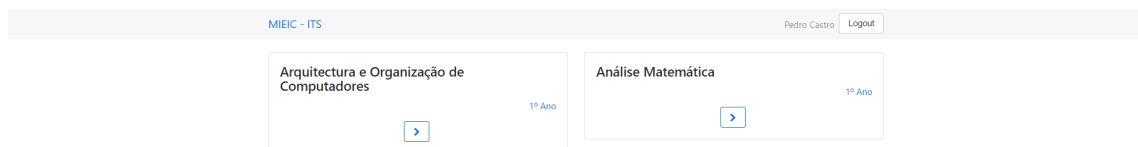


Figura 4.4: Estudante validado

- (c) Um utilizador que seja um gestor do sistema, tem duas secções no seu *dashboard*: uma secção com uma listagem dos seus grupos e uma listagem de todos os grupos que gere - este último caso é visível na figura 4.5.

Solução Desenvolvida

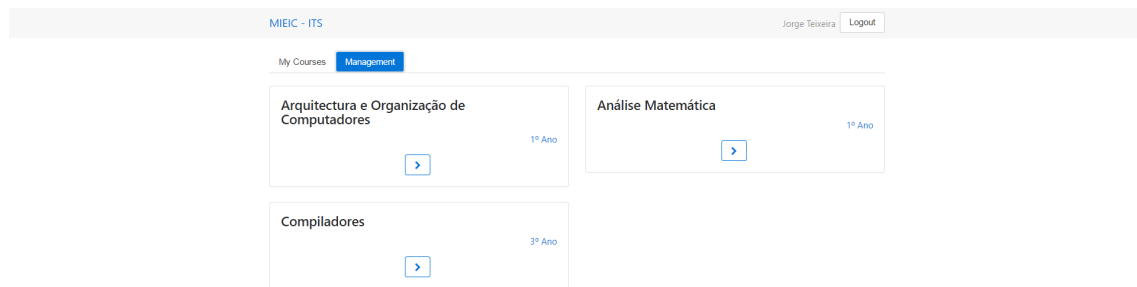


Figura 4.5: Gestor validado

- (d) Um utilizador que seja administrador do sistema, tem as mesmas secções referidas anteriormente, mas são acrescentadas algumas vistas que são de administração geral do sistema. Numa destas secções exclusivas é possível ver as informações gerais de todos os grupos existentes no sistema, com um sistema de semáforos que avalia os grupos que constituem o sistema - figura 4.6. Este sistema de semáforos foi elaborado para permitir uma maior facilidade na análise do estado dos vários grupos que constituem o sistema. A configuração do semáforo é personalizável, consoante as preferências do administrador. Sendo assim, o estado vermelho de um grupo, significa para o administrador, que este necessita de uma atenção adicional, um grupo de estado amarelo, tem um nível intermédio de alerta e um grupo com estado verde não se encontra em situação problemática. Para além do semáforo e da sua configuração é possível ver os cinco mais recentes *issues* abertos e o utilizador tem a opção de criar um novo grupo, ou adicionar uma nova unidade curricular - figura 4.7.

Solução Desenvolvida

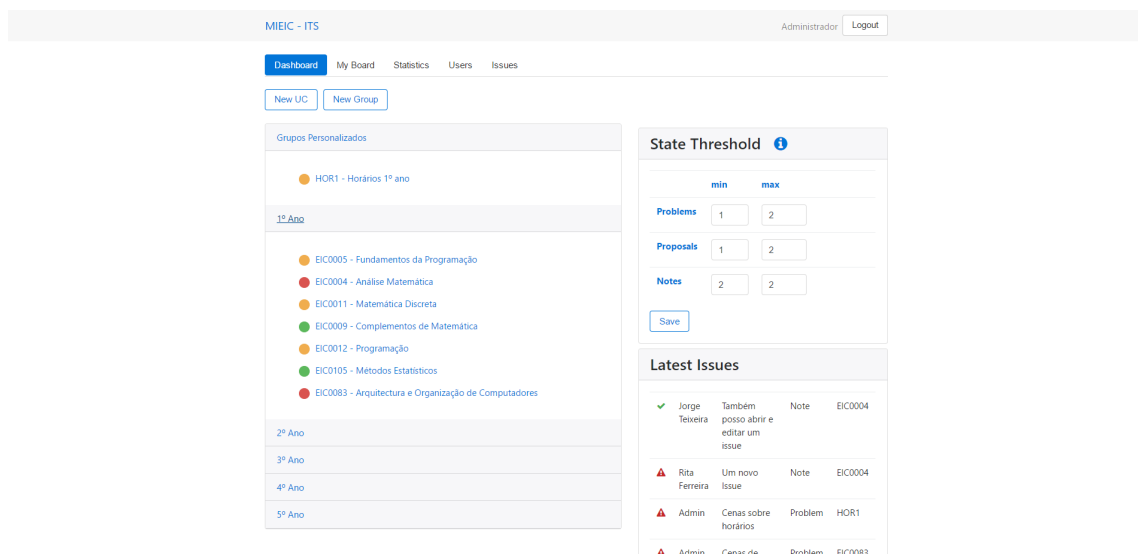


Figura 4.6: *Dashboard* de administrador

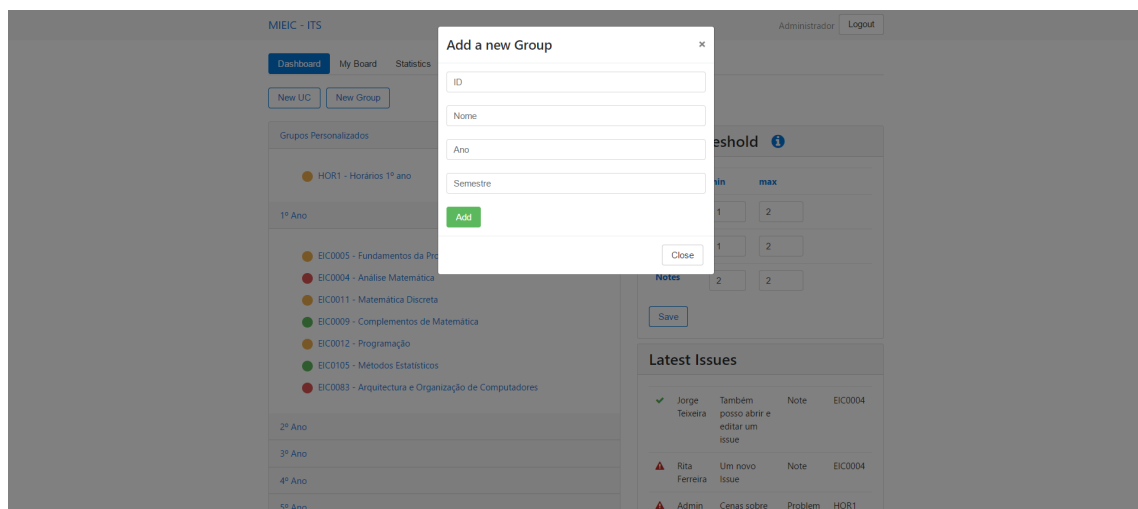


Figura 4.7: Criação de um novo grupo

- (e) Um administrador tem também uma secção para ver as estatísticas mais importantes do sistema. Nesta página podemos ver uma distribuição dos *issues* abertos do sistema por ano curricular, o número de *issues* abertos no sistema, o grupo com mais *issues* abertos, uma distribuição mensal dos *issues* totais e uma distribuição dos tipos de *issues* existentes - visível na figura 4.8.

Solução Desenvolvida

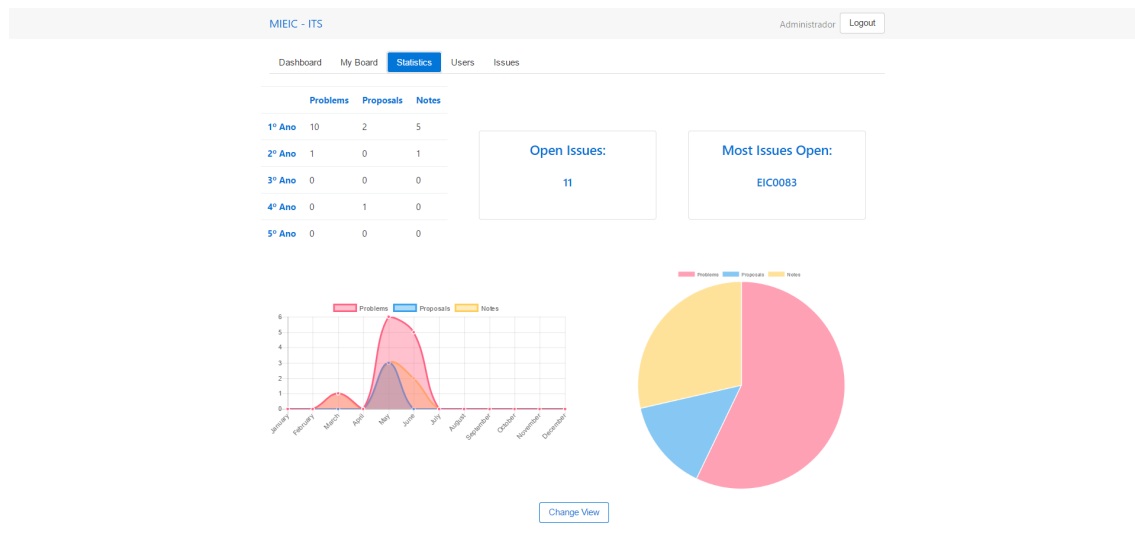


Figura 4.8: Página de estatísticas do sistema

- (f) Um administrador tem ao seu dispor uma listagem de todos os utilizadores do sistema, onde pode eliminar utilizadores já validados e validar utilizadores recém registrados - ver figura 4.9.

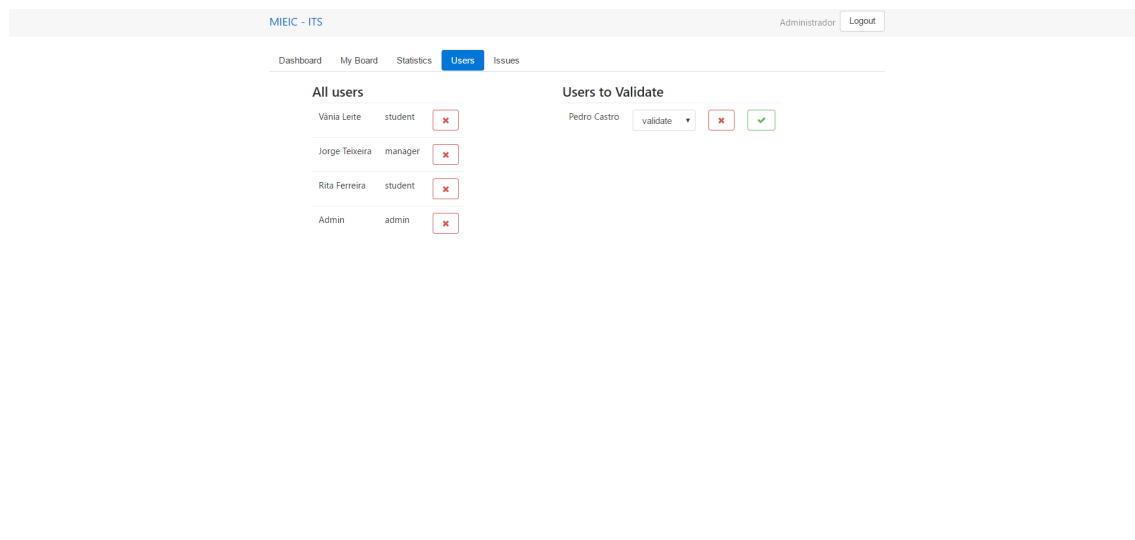


Figura 4.9: Listagem de utilizadores do sistema

- (g) Um administrador tem também uma secção onde pode fazer a gestão de todos os *issues* existentes no sistema - ver figura 4.10

Solução Desenvolvida

MIEIC - ITS

AdministradorLogout

DashboardMy BoardStatisticsUsersIssues

All Issues

Jorge Teixeira	Primeiro Issue	Solved	EIC0004	Problem	Mar 18, 2017, 11:22:14 PM	X
Jorge Teixeira	Primeiro	Open	EIC0083	Note	Mar 31, 2017, 9:12:24 PM	X
Jorge	Primeiro	On Going	EIC0023	Problem	May 16, 2017, 6:18:43 PM	X
Jorge	Segundo	Solved	EIC0023	Note	May 16, 2017, 6:19:12 PM	X
Jorge	Novo Issue	Solved	EIC0083	Problem	May 16, 2017, 6:46:42 PM	X
Jorge	Um Issue	On Going	EIC0011	Problem	May 16, 2017, 6:47:08 PM	X
Jorge	Número 1	Open	EIC0012	Problem	May 16, 2017, 7:00:37 PM	X
Jorge	Teste	On Going	EIC0083	Problem	May 25, 2017, 11:57:59 AM	X
Jorge	Novo	Open	EIC0083	Problem	May 25, 2017, 12:22:24 PM	X
Jorge	Primeiro da UC	On Going	EIC0053	Proposal	May 25, 2017, 2:26:21 PM	X
admin	Mais um	Open	EIC0083	Proposal	May 25, 2017, 2:27:41 PM	X
admin	Teste1	On Going	EIC0083	Note	May 25, 2017, 2:32:22 PM	X
Admin	Um	Open	EIC0005	Proposal	May 25, 2017, 11:53:55 PM	X

Figura 4.10: Listagem de problemas do sistema

3. Depois de aceder ao seu *dashboard*, um utilizador poderá visualizar a página de um grupo. Nesta página é possível ver a informação geral do grupo e uma listagem de todos os *issues* existentes neste grupo. Esta listagem permite pesquisas de *issues* por nome do *issue* ou pelas *tags* que estes possam ter. É possível também filtrar os *issues* por tipo e pelo seu estado. Sendo assim, conseguimos ter acesso aos *issues* abertos, ou aos já fechados, permitindo-nos ter acesso a um historial de *issues* - ver figura 4.11. Esta página tem duas variações de visualização:

MIEIC - ITS

AdministradorLogout

< Arquitectura e Organização de Computadores

Add MembersAdd ManagerNew Issue

▲

⚠

✓

Problem

Proposal

Note

All

Search by description

Search by tag

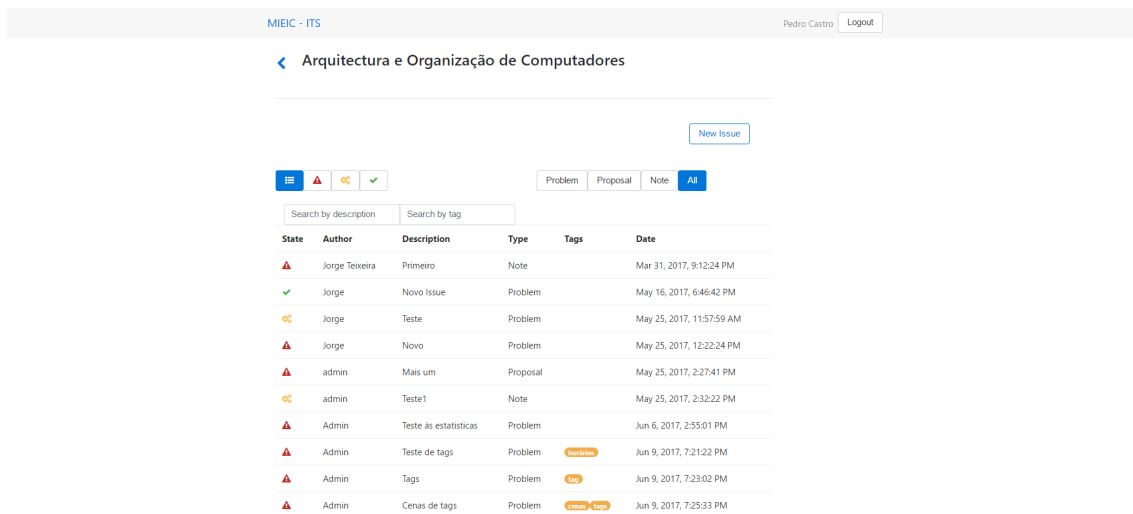
State	Author	Description	Type	Tags	Date
▲	Jorge Teixeira	Primeiro	Note	Mar 31, 2017, 9:12:24 PM	
✓	Jorge	Novo Issue	Problem	May 16, 2017, 6:46:42 PM	
⚠	Jorge	Teste	Problem	May 25, 2017, 11:57:59 AM	
▲	Jorge	Novo	Problem	May 25, 2017, 12:22:24 PM	
▲	admin	Mais um	Proposal	May 25, 2017, 2:27:41 PM	
⚠	admin	Teste1	Note	May 25, 2017, 2:32:22 PM	
▲	Admin	Teste às estatísticas	Problem	Jun 6, 2017, 2:55:01 PM	
▲	Admin	Teste de tags	Problem	Jun 9, 2017, 7:21:22 PM	🔖
▲	Admin	Tags	Problem	Jun 9, 2017, 7:23:02 PM	🔖
▲	Admin	Cenas de tags	Problem	Jun 9, 2017, 7:25:33 PM	🔖🔖

Figura 4.11: Página de um grupo - Administrador

- (a) Na primeira, um utilizador (sem nível de gestão ou um gestor do grupo), tem acesso apenas às informações descritas anteriormente e ao formulário de inserção de um novo

Solução Desenvolvida

issue, como é possível ver na figura 4.12.



State	Author	Description	Type	Tags	Date
⚠	Jorge Teixeira	Primeiro	Note		Mar 31, 2017, 9:12:24 PM
✓	Jorge	Novo Issue	Problem		May 16, 2017, 6:46:42 PM
⚠	Jorge	Teste	Problem		May 25, 2017, 11:57:59 AM
⚠	Jorge	Novo	Problem		May 25, 2017, 12:32:24 PM
⚠	admin	Mais um	Proposal		May 25, 2017, 2:27:41 PM
⚠	admin	Teste1	Note		May 25, 2017, 2:32:22 PM
⚠	Admin	Teste às estatísticas	Problem		Jun 6, 2017, 2:55:01 PM
⚠	Admin	Teste de tags	Problem	⚠	Jun 9, 2017, 7:21:22 PM
⚠	Admin	Tags	Problem	⚠	Jun 9, 2017, 7:23:02 PM
⚠	Admin	Cenas de tags	Problem	⚠	Jun 9, 2017, 7:25:33 PM

Figura 4.12: Página de um grupo - Estudante

- (b) Na segunda, um administrador, tem possibilidade de dar permissões de visualização a novos utilizadores e a possibilidade de dar permissões de gestão a um utilizador, já definido como gestor do sistema, como é visível na figura 4.13.

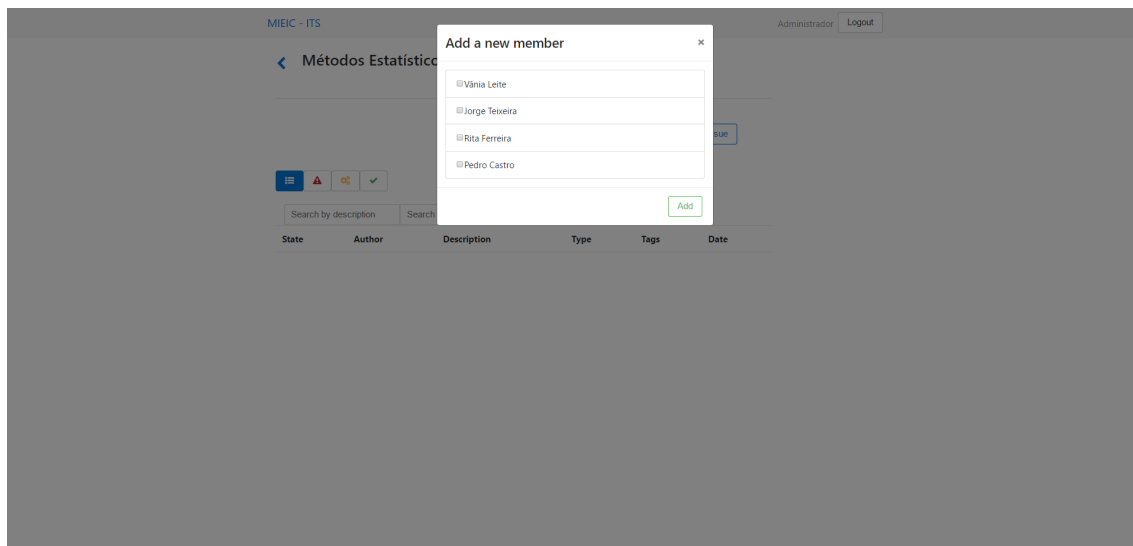


Figura 4.13: Alterar permissões de um utilizador

4. Para ter acesso à página de um *issue*, é necessário que o utilizador tenha permissões de gestão do grupo a que corresponde este *issue*, ou que este seja um administrador do sistema. Nesta página, são visíveis as informações gerais deste *issue* e todos os comentários que este *issue* tem, ordenados cronologicamente. Um *issue* que tenha o estado fechado, não tem a

Solução Desenvolvida

opção de adicionar novos comentários, como representado na figura 4.14. Caso o utilizador que esteja a ver a página do *issue* tenha sido o seu criador, ou seja um administrador, consegue proceder à edição da informação aí presente, caso contrário, esta opção fica escondida. Procedendo a esta edição é então permitido ao utilizador que feche o *issue* em questão e podemos ver esta opção na figura A.2.

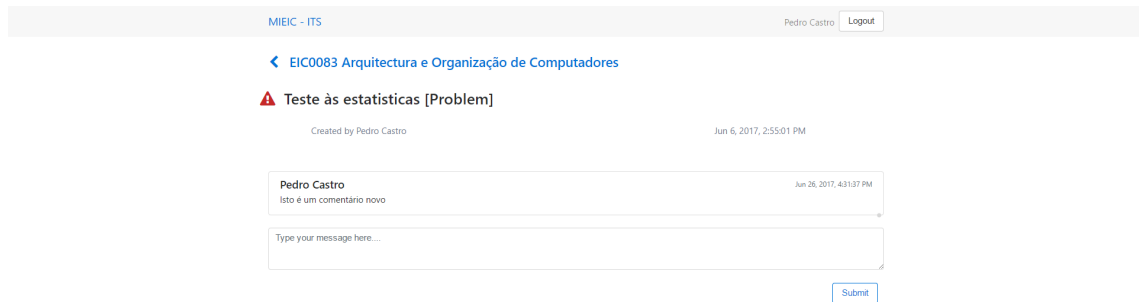


Figura 4.14: Página de um problema - Estudante

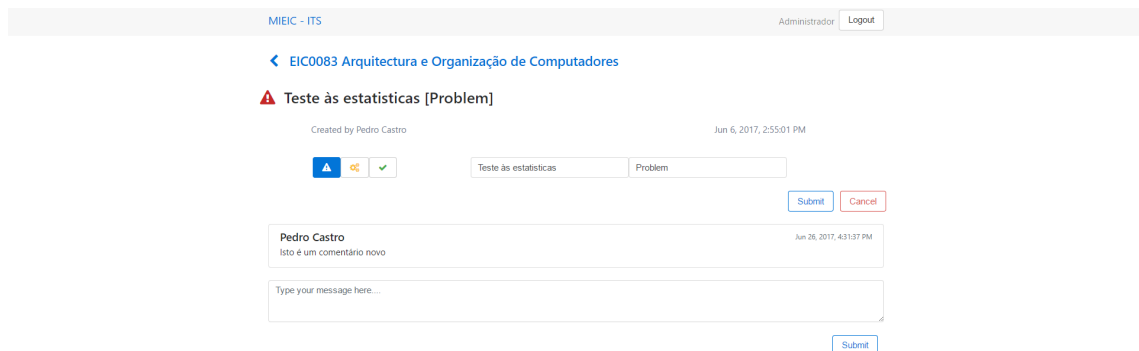


Figura 4.15: Página de um problema - Administrador

Solução Desenvolvida

Capítulo 5

Testes e resultados obtidos

Como já veio a ser referido, um dos grandes propósitos do trabalho realizado no âmbito desta dissertação, é a possibilidade de no final do trabalho de desenvolvimento, poder pôr em prática a utilização desta ferramenta. Sendo assim, é de grande importância que se proceda a testes de usabilidade e validação para que seja possível observar problemas ou elementos em falta que o sistema possa ter e em que a sua adição possa trazer um grande benefício para a finalização do sistema.

Tendo em conta que numa primeira fase, o ambiente de utilização desta ferramenta, contará quase em exclusivo apenas com elementos da comissão de acompanhamento do MIEIC, é assim pertinente que alguns destes elementos pudessem fazer testes com casos de uso reais, de forma a podermos aferir se existem melhorias a fazer, se aquilo que foi desenvolvido vai de encontro aquilo que era desejável e qual é a sua opinião em relação a uma série de critérios de validação, que serão apresentados de seguida.

5.1 Metodologia dos testes realizados

Os testes que foram realizados tiveram como principal foco, as funcionalidades que foram desenvolvidas para a plataforma e a análise dos fluxos de utilização destas várias funcionalidades.

Inicialmente foi explicado o propósito do projeto e as funcionalidades que este inclui. Posteriormente, foi pedido aos membros da comissão de acompanhamento que fizessem uso de todas as funcionalidades que pertencem aos papéis de estudantes e gestores do sistema. Em paralelo, foi pedido que fizessem uma avaliação de vários pontos e métricas previamente definidas, de forma a termos parâmetros de comparação entre todas as opiniões. Os pontos que foram pedidos aos membros da comissão de acompanhamento para preencher foram:

Sistema atual vs Sistema proposto

Estes termos de avaliação avaliam em termos comparativos o sistema tal como existe no momento atual no contexto do acompanhamento curricular e o sistema que é proposto ser implementado neste trabalho. E os parâmetros que foram identificados são:

- Parâmetros gerais
 - Facilidade para reportar problemas
 - Facilidade para obter opiniões de vários elementos
 - Facilidade para solucionar problemas
 - Organização dos problemas por grupos
- Parâmetros referentes a estudantes
 - Facilidade para reportar problemas
 - Grau de satisfação ao ver problemas serem solucionados
 - Nível de envolvimento na resolução de problemas
 - Nível de confiança no levantamento de problemas
- Parâmetros referentes a gestores
 - Facilidade na resolução de problemas
 - Organização dos problemas por grupos
 - Facilidade para recolher opiniões de vários elementos
 - Autonomia para resolver problemas
 - Nível de confiança para resolver problemas
- Parâmetros referentes a administradores
 - Manter-se atualizado quanto aos problemas abertos
 - Recolher informações oportunas através de estatísticas
 - Facilidade para resolver problemas
 - Confiança na resolução de problemas por parte de outros
 - Facilidade na gestão do sistema

Avaliação das funcionalidades desenvolvidas

Para este campo de avaliação foi pedido que os elementos que realizaram os testes avaliassem de forma independente a utilidade das várias funcionalidades que foram desenvolvidas para a plataforma, seguindo os seguintes pontos:

- Funcionalidades referentes a estudantes

Testes e resultados obtidos

- *Dashboard* pessoal do utilizador
- Listagem de problemas
- Reportar um problema
- Adicionar *tags* aos problemas adicionados
- Pesquisa de problemas por nome
- Pesquisa de problemas por *tags*
- Funcionalidades referentes a gestores
 - Listagem de grupos que um utilizador gere
 - Listagem de problemas
 - Reportar um problema
 - Adicionar *tags* aos problemas adicionados
 - Pesquisa de problemas por nome
 - Pesquisa de problemas por *tags*
 - Ver informação de um problema na sua página própria
 - Adicionar comentários a um problema
- Funcionalidades referentes a administradores
 - Sistema de semáforo no *dashboard* pessoal
 - Ver problemas adicionados recentemente
 - Configurar o sistema de semáforos
 - Listagem de todos problemas
 - Listagem de todos os utilizadores
 - Validar utilizadores
 - Resolver um problema

Tanto os parâmetros comparativos entre o sistema atual e o proposto, como os parâmetros de avaliação de funcionalidades, receberam uma avaliação de 1 a 5. No caso dos parâmetros comparativos, a avaliação significa:

1. Muito mau
2. Mau
3. Razoável
4. Bom
5. Muito bom

E no que toca aos parâmetros de avaliação de funcionalidades, as avaliações correspondem a:

1. Inútil
2. Pouco útil
3. Mais valia
4. Útil
5. Muito útil

5.2 Resultados e observações

Visto os testes terem sido realizados por alunos pertencentes à comissão de acompanhamento do MIEIC, as funcionalidades que são destinadas a administradores, apesar de terem sido sujeitas a testes, podem não ser totalmente esclarecedoras no que toca ao benefício que estas funcionalidades podem trazer a um administrador. No entanto, é importante analisarmos os comentários e resultados que obtivemos nestes pontos de avaliação, tendo em conta que algumas das funcionalidades exclusivas de administradores, podem ser consideradas importantes para gestores ou até mesmo estudantes e também porque apesar de não termos notas de elementos que efetivamente pudessem fazer uso de algumas destas funcionalidades, devido ao núcleo próximo de interação entre todos os membros da comissão de acompanhamento (tanto alunos como professores), a avaliação dos primeiros pode tornar-se importante e próxima da realidade.

5.2.1 Resultados referentes à comparação entre o sistema atual e o proposto

Tal como referido anteriormente, os elementos sujeitos aos testes realizados, avaliaram os vários pontos numa métrica de 1 a 5, sendo 5 em todos os casos a nota mais positiva e 1 a mais negativa. Na análise destas avaliações, estas notas foram somadas, tanto para as notas referentes ao sistema proposto, como as referentes ao sistema atual e foi finalmente feita a diferença entre os dois sistemas, pela ordem referida. Desta forma uma diferença positiva implica uma apreciação mais benéfica para o sistema proposto e uma diferença negativa implica que o sistema atual sai em vantagem numa comparação direta, para cada ponto analisado. Com este método de análise, conseguimos ver quais são os pontos em que o sistema proposto sai em vantagem em relação ao sistema atual e vice-versa. E temos também uma visão de qual é a importância que cada elemento confere aos vários pontos em análise, sendo que quanto maior for a diferença obtida, maior será o benefício ou prejuízo que uma mudança de sistemas trará.

No que refere aos parâmetros gerais discriminados na secção anterior, conseguimos perceber que todos os elementos interpretam o sistema proposto como uma melhoria relativamente ao sistema que existe de momento, sendo que todos os pontos referentes obtêm uma diferença positiva, na análise final dos resultados. Existe no entanto, um ponto que sobressai em relação aos seus

pares, visto que os elementos que realizaram a avaliação interpretam que o sistema proposto traz uma organização aos problemas levantados bastante melhor que o sistema atual.

No que toca à visão que os estudantes podem ter do sistema, existem pontos mais divergentes. Dos quatros pontos avaliados pelos membros da comissão de acompanhamento, três obtiveram uma diferença positiva, apesar de ser pouco significativo. Para além destes pontos, aquele que se refere ao nível de confiança que o sistema proposto traz a um estudante que levante um problema, obteve uma avaliação negativa, apesar de ser também pouco expressivo.

Nos pontos referentes ao ponto de vista de um gestor - posição que deve ser maioritariamente ocupada por elementos das comissões de acompanhamento, tal como os elementos que realizaram os testes - conseguimos obter uma totalidade de resultados positivos. O ponto que se destacou pela positiva desta análise final foi, mais uma vez, a organização que o sistema traz como novidade à gestão de problemas, vantagem esta que é mais observada para posições de gestão. O ponto com piores resultados (apesar de positivos) refere-se ao nível de confiança que um gestor tem aquando da resolução de um problema. Os pontos que se referem à visão dos administradores do sistema, obtiveram uma totalidade de observações positivas, sendo que os pontos que se destacam mais pela positiva são a possibilidade de ter expostas algumas estatísticas, que podem facilitar a gestão do sistema e a facilidade que o sistema proposto traz para a gestão do acompanhamento curricular. O ponto que obteve uma diferença menos positiva na generalidade das avaliações, refere-se ao aumento de confiança do administrador na resolução de problemas por parte de outros.

5.2.2 Resultados referentes à avaliação das funcionalidades desenvolvidas

Para estes testes de avaliação das funcionalidades desenvolvidas, foi pedido aos elementos que realizaram os testes que avaliassem de 1 a 5 os vários pontos que foram já referidos na secção anterior.

Para proceder à análise destes resultados, as notas de todos os elementos foram somadas e no fim esse resultado foi dividido pelo número de elementos que realizaram os testes, de forma a normalizar as respostas. Desta forma, uma nota abaixo de 3, representa uma nota insuficiente e superior a 3 representa uma nota positiva. Neste campo, todas as funcionalidades que foram avaliadas obtiveram uma nota final positiva. No entanto, existem alguns casos de destaque em cada um dos pontos de vista avaliados - estudante, gestor e administrador -, sendo um deles a funcionalidade de um estudante poder reportar um problema através do sistema. Ainda no ponto de vista dos estudantes, a possibilidade de estes terem acesso a uma listagem dos problemas que cada grupo a que pertencem tem, também foi considerado bastante positivo.

No campo referente aos gestores do sistema, para além das funcionalidades destacadas anteriormente, também foi considerada uma mais valia a possibilidade de problemas receberem comentários por parte dos gestores. Por fim, duas das funcionalidades que os administradores do sistema têm ao seu dispor, são alvos de destaque. O primeiro pela positiva, sendo a funcionalidade com maior avaliação final, que é a possibilidade de um administrador poder ver os problemas mais recentes a serem reportados. A funcionalidade que obteve a posição menos positiva deste grupo - e também da generalidade dos testes - é a possibilidade de configurar o semáforo de administrador.

5.3 Conclusões

Em geral, os testes resultaram em observações positivas no sentido da implementação do sistema proposto, tendo em vista que os resultados dos testes realizados retornaram pontuações positivas e uma clara preferência pela forma como o sistema proposto resolve os problemas que existem no acompanhamento curricular, em detrimento do sistema atual.

Em resumo, os resultados que os testes produziram, dizem-nos que a generalidade das funcionalidades implementadas traz uma vantagem, ou em último caso não traz um retrocesso, em relação ao que existe de momento no acompanhamento curricular. Neste campo, as funcionalidades que foram consideradas mais positivas neste sistema foram a possibilidade de reportar um problema, muito por causa da facilidade que o sistema traz a esta tarefa. Outra tarefa que sai beneficiada, aos olhos dos elementos que testaram a plataforma, é a facilidade com que um administrador consegue manter-se a par dos problemas que vão sendo reportados, com a opção de ver os problemas mais recentes.

Ainda fazendo uma conclusão dos resultados dos testes, podemos ver que em termos comparativos, o sistema proposto obtém uma melhor projeção do que o sistema atual. Neste capítulo, sabemos que os parâmetros em que a melhoria comparativa entre o sistema atual e o proposto é maior para este último é no que toca à organização dos problemas por grupos, que é visto como uma grande vantagem do sistema proposto e que facilita em grande medida o acompanhamento curricular e a resolução de problemas. Para além disto, a existência de estatísticas dedicadas à gestão do sistema, também é visto como uma vantagem do novo sistema.

Por outro lado, como principal ponto negativo, sabemos que é com alguma relutância que os elementos que realizaram os testes, vêem a possibilidade de estudantes fazerem o levantamento de problemas, sem se sentirem de alguma forma mais expostos. Isto deve-se em grande medida ao facto de todos os problemas terem um utilizador a si associados como o seu autor e isso pode tornar-se um facto que possa inibir os estudantes a reportar um problema no sistema. Porém, é possível ultrapassar esta situação se estes problemas levantados por estudantes, forem filtrados por alunos que façam a gestão do sistema. Desta forma, antes dos problemas serem visíveis por administradores do sistema, ou por professores que façam parte do sistema, podem ser visionados por outros estudantes gestores, sendo que desta forma os problemas podem ser filtrados por importância e validados antes de serem visíveis para todos. Para além disto ser uma solução para o problema da falta de confiança para reportar um problema por parte dos estudantes, é também uma forma de reduzir problemas replicados, reduzir o número de problemas presentes a administradores e é também um processo que se assemelha ao método de levantamento de problemas atual, em que um estudante pode reportar problemas a membros da comissão de acompanhamento, antes destes chegarem a professores ou ao diretor de curso.

Alguns outros pontos que não obtiveram um destaque pela positiva, tal como referido anteriormente, encontram-se em exclusivo no ponto de vista de gestão e administração do sistema. Para os elementos da comissão de acompanhamento, a facilidade com que um gestor resolve um problema, o nível de confiança que este tem para tratar de um problema, a facilidade que um administrador se mantém a par dos problemas existentes e a confiança que este tem que outros possam resolver os problemas existentes, são os pontos avaliados que não obtiveram um grande destaque nos resultados obtidos. Apesar de todos terem tido avaliações positivas, estas não foram muito expressivas e isto pode ter que ver com o facto do sistema proposto ter bastantes semelhanças, no que toca ao método de resolução de problemas, em comparação com o sistema atual. Ou seja, apesar de ter sido evidente que um sistema informatizado traga algumas vantagens no que concerne a estes parâmetros, não existe uma discrepância acentuada entre os dois sistemas.

Para além das pontuações que foram obtidas pelas avaliações dos testes, a opinião dos elementos que os realizaram, refletiu sempre uma vontade de utilizar o sistema. Para além das reações positivas, a experiência de utilização do sistema deu origem a várias propostas de melhoria e sugestões de como utilizar aquilo que foi desenvolvido em prol da facilidade e efetividade da utilização do sistema e para ir de encontro aquilo que deve ser o acompanhamento curricular idealmente.

As propostas de melhoria que foram feitas pelos membros da comissão de acompanhamento resultam da experiência que tiveram durante a realização dos testes e referem-se em grande parte a melhorias que podem ser implementadas, de forma a facilitar o acompanhamento curricular nos vários papéis que os utilizadores do sistema podem ter.

A proposta que foi mais referida pelos membros da comissão de acompanhamento, foi o alargamento da funcionalidade de semáforos para os gestores, não se ficando pelo papel de administrador do sistema. Este semáforo não seria completo, mas conteria os grupos que estes gestores tivessem acesso. Esta proposta foi feita no sentido de aumentar a organização dos grupos e por sua vez, dos problemas que estes devem gerir. Algumas das propostas mais referidas assentam no sistema de comentários a problemas. Uma das propostas passa por introduzir um sistema de *upvotes* aos comentários, sendo que desta forma, torna-se mais visual quais são as propostas de resolução de problemas mais populares pelos gestores. Outra sugestão referente à área de comentários, era a introdução de *threads* de comentários, visto que desta forma seria mais fácil seguir um raciocínio dentro dos comentários, sem que o fluxo normal de comentários possa tornar mais caótica a tarefa de solucionar um problema.

Uma última sugestão proposta, passa por aumentar o nível de notificações que é acionada no sistema. Estas notificações podem servir para os gestores se manterem a par de um problema que identifiquem como importante, ou por exemplo, receberem um alerta sempre que fosse adicionado um novo problema a um grupo que gerem.

Concluindo, o sistema parece trazer algum consenso no que toca à sua mais valia principal,

Testes e resultados obtidos

que é a organização e facilidade de resolução dos problemas. Podemos notar que existem alguns pontos a melhorar ou expandir, no entanto, na generalidade da plataforma, podemos ver que existe uma melhoria em relação ao sistema utilizado atualmente. Tomando partido das várias sugestões propostas pelos membros da comissão de acompanhamento, é possível chegar a um sistema perto do ideal e aumentar a produtividade do acompanhamento curricular.

Capítulo 6

Conclusões

Durante o desenvolvimento deste trabalho, conseguimos ficar com uma visão mais clara daquilo que representa o acompanhamento curricular, numa instituição de ensino superior como é o caso da FEUP. Sabemos que esta gestão de problemas e situações de risco é de grande importância para a manutenção da qualidade da instituição e promove uma melhoria no funcionamento diário e no ambiente que é vivido.

Sabemos também que para ser possível alcançar este estado de qualidade, a FEUP serve-se da comissão de acompanhamento como elo de ligação entre os alunos, os professores e os membros do secretariado e administração da FEUP. Este órgão é de grande importância para a constante manutenção e resolução de problemas que vão sendo reportados no dia a dia da faculdade. A esta comissão compete o trabalho de manter-se a par das situações problemáticas que vão acontecendo nos seus respetivos cursos e procurar solucionar estas situações, ou então encaminhá-las para as pessoas competentes.

Para percebermos de que forma podemos agir para melhorar este sistema, foi necessário perceber que tecnologias são utilizadas atualmente para proceder à resolução de problemas e quais são as plataformas que os sistemas de ensino implementam nos dias de hoje. Assim ficamos a conhecer os *issue tracking system*, que servem essencialmente para fazer o acompanhamento dos problemas que os vários utilizadores destes sistemas vão reportando. Este método torna-se bastante eficaz em equipas de trabalho grandes e torna o processo de recolha de problemas e de solucionamento dos mesmos, bastante acelerado e eficiente. Também pudemos observar o funcionamento de alguns *learning management systems*, tal como o *Moodle*, que possibilita aos professores e estudantes que partilhem uma série de elementos digitais, que facilitam o ensino e aumentam o acompanhamento que os membros dessas instituições podem ter, usando uma plataforma destas.

Analisando aquilo que se faz de melhor, nas áreas que foram mencionadas anteriormente, chegamos à conclusão que apesar de nenhum destes sistemas servir inteiramente para solucionar o problema que temos inicialmente, conseguimos perceber que utilizando a lógica de ambos, é possível desenvolver um sistema que possa, por um lado fazer a recolha e tratar da resolução

Conclusões

dos vários problemas que vão sendo levantados e por outro lado, possa incluir tanto professores como estudantes num único sistema, aumentando a eficiência e a qualidade do acompanhamento curricular.

Neste trabalho, foi importante percebermos de que forma o sistema em funcionamento é processado e quais são os grandes pontos positivos e negativos daquilo que é feito. Só assim foi possível traçar um caminho de desenvolvimento de um sistema, que apesar de informatizado, procura não descaracterizar aquilo que é o funcionamento do acompanhamento curricular, tal como ele é atualmente. Sendo assim, foi possível discriminar os pontos que deveriam ser mantidos e quais poderiam ser melhorados num sistema novo.

As principais características que foram traçadas à partida para o desenvolvimento desta plataforma, foram imputar uma grande rapidez tanto ao levantamento de problemas, como à sua resolução. Para além disso, uma restrição inicial deste sistema era incluir tanto professores como estudantes, de forma a que através de uma única plataforma fosse possível englobar todos os problemas que fossem sendo identificados. Uma melhoria que foi inicialmente planeada com a criação de um sistema informático, foi a possibilidade de tornar o acompanhamento curricular mais organizado, no que toca à disposição de problemas, podendo estes estar agrupados de forma intuitiva e prática, sem que se torna-se caótico a resolução dos mesmos.

Ao proceder à validação do sistema desenvolvido, através de testes efetuados com a participação dos elementos da comissão de acompanhamento do MIEIC, conseguimos perceber que o propósito do sistema foi na sua generalidade alcançado. Foram referidos vários pontos, em que a criação de um sistema informático, torna o acompanhamento curricular mais eficiente e mais rápido. No entanto, existem algumas desvantagens que este sistema pode vir a trazer com uma utilização mais frequente do sistema. Mesmo assim, procedendo a alguns melhoramentos no futuro, será possível solucionar em grande parte estes problemas e tornar o sistema mais próximo do esperado e desejado.

Neste seguimento, é importante referir que o sistema deve ser sujeito a melhoramentos no futuro, não só para se adaptar às constantes mudanças a que uma instituição de ensino está sempre sujeita, mas também para permitir que o sistema se molde às mais importantes necessidades que forem sendo identificadas pelos seus utilizadores. O sistema foi construído, tendo em mente que estas mudanças serão sempre necessárias e por isso mesmo, será fácil proceder a melhorias e aumentos do comportamento atual do sistema. Podem ser acrescentadas várias funcionalidades que tragam uma vantagem prática e visível ao acompanhamento curricular da FEUP no futuro, mas que também facilitem o dia a dia dos membros da sua comunidade.

Em suma, podemos referir que a proposta inicial desta dissertação foi concluída, tendo sido

Conclusões

implementadas as várias funcionalidades que permitem exercer o acompanhamento curricular por parte dos vários intervenientes da comunidade FEUP. Tal como previsto inicialmente, esta plataforma deverá ser utilizada no decorrer do próximo ano letivo, no entanto, será utilizada por um grupo de elementos mais restrito numa fase inicial. Isto deve-se à necessidade de promover testes de utilização mais extensos e que permitam avaliar a real eficiência que o sistema traz. Sendo assim, a comissão de acompanhamento fará uso da plataforma nesta fase inicial e assim que o sistema esteja completamente validado e adaptado às necessidades da comunidade, então poderá proceder-se à propagação do sistema por todos os membros do DEI e numa fase posterior, da FEUP.

Conclusões

Referências

- [AAZ08] A. Al-Ajlan e H. Zedan. Why Moodle. *2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pages 58–64, 2008. doi:10.1109/FTDCS.2008.22.
- [AHS14] Karan Aggarwal, Abram Hindle e Eleni Stroulia. Co-evolution of project documentation and popularity within github. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, pages 360–363, New York, NY, USA, 2014. ACM. URL: <http://doi.acm.org/10.1145/2597073.2597120>, doi:10.1145/2597073.2597120.
- [anga] Github angular. <https://github.com/angular/angular>. Last Accessed: 2017-06-19.
- [angb] The ultimate angular cli reference guide. <https://www.sitepoint.com/ultimate-angular-cli-reference/>. Last Accessed: 2017-06-19.
- [BLJ⁺13] T. F. Bissyandé, D. Lo, L. Jiang, L. Réveillère, J. Klein e Y. L. Traon. Got issues? who cares about it? a large scale investigation of issue trackers from github. In *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*, pages 188–197, Nov 2013. doi:10.1109/ISSRE.2013.6698918.
- [boo] Twitter, Inc bootstrap. <http://getbootstrap.com/>. Last Accessed: 2017-06-19.
- [Cho13] Kristina Chodorow. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. "O'Reilly Media, Inc.", 2013.
- [exp] Node.js Foundation express. <https://expressjs.com/>. Last Accessed: 2017-06-19.
- [Fla06] David Flanagan. *JavaScript: the definitive guide*. "O'Reilly Media, Inc.", 2006.
- [GSHH16] Adrian German, Santiago Salmeron, Wonyong Ha e Bo Henderson. Mean web development: A tutorial for educators. In *Proceedings of the 17th Annual Conference on Information Technology Education*, SIGITE '16, pages 128–129, New York, NY, USA, 2016. ACM. URL: <http://doi.acm.org/10.1145/2978192.2978247>, doi:10.1145/2978192.2978247.
- [JD03] J. N. Johnson e P. F. Dubois. Issue tracking. *Computing in Science Engineering*, 5(6):71–77, Nov 2003. doi:10.1109/MCISE.2003.1238707.
- [KGD11] Sheo Kumar, Anil Kumar Gankotiya e Kamlesh Dutta. A comparative study of moodle with other e-learning systems. In *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology*, volume 5, pages 414–418, 2011. doi:10.1109/ICECTECH.2011.5942032.

REFERÊNCIAS

- [KMAC08] Gorgi Kakasevski, Martin Mihajlov, Sime Arsenovski e Slavcho Chungurski. Evaluating usability in learning management system moodle. In *Proceedings of the International Conference on Information Technology Interfaces, ITI*, pages 613–618, 2008. doi:10.1109/ITI.2008.4588480.
- [KPTD11] Andreas Konstantinidis, Pantelis M. Papadopoulos, Thrasyvoulos\ Tsiatsos e Stavros Demetriadis. Selecting and Evaluating a Learning Management System: A Moodle Evaluation Based on Instructors and Students. *International Journal of Distance Education Technologies*, 9(3):13–30, 2011. URL: <http://www.igi-global.com/article/selecting-evaluating-learning-management-system/55796>, doi:10.4018/jdet.2011070102.
- [LMK15] Pascal Lessel, Marc Müller e Antonio Krüger. Towards a novel issue tracking system for "industry 4.0" environments. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '15*, pages 1809–1814, New York, NY, USA, 2015. ACM. URL: <http://doi.acm.org/10.1145/2702613.2732720>, doi:10.1145/2702613.2732720.
- [mla] Heroku Dev Center mlab. <https://devcenter.heroku.com/articles/mongolab>. Last Accessed: 2017-06-19.
- [nod] Node.js Foundation node.js. <https://nodejs.org/en/>. Last Accessed: 2017-06-19.
- [ODA⁺15] Marco Ortu, Giuseppe Destefanis, Bram Adams, Alessandro Murgia, Michele Marchesi e Roberto Tonelli. The jira repository dataset: Understanding social aspects of software development. In *Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering, PROMISE '15*, pages 1:1–1:4, New York, NY, USA, 2015. ACM. URL: <http://doi.acm.org/10.1145/2810146.2810147>, doi:10.1145/2810146.2810147.
- [PJC15] A. J. Poulter, S. J. Johnston e S. J. Cox. Using the mean stack to implement a restful service for an internet of things application. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 280–285, Dec 2015. doi:10.1109/WF-IoT.2015.7389066.
- [SC05] N. Serrano e I. Ciordia. Bugzilla, itracker, and other bug trackers. *IEEE Software*, 22(2):11–13, March 2005. doi:10.1109/MS.2005.32.
- [typ] Microsoft typescript. <https://github.com/Microsoft/TypeScript>. Last Accessed: 2017-06-19.

Anexo A

Resultados obtidos nos testes realizados

	Sistema Atual	Sistema Proposto	Sistema Atual	Sistema Proposto	Sistema Atual	Sistema Proposto	Sistema Atual	Sistema Proposto	Sistema Atual	Sistema Proposto	Diferença	
Global	Elaborado 1	Elaborado 2	Elaborado 3	Elaborado 3	Elaborado 3	Elaborado 3	Total	Total				1 Muito mau
Facilidade para reportar um problema	3	3	3	4	5	3	4	5	14	12	5	3 Razoável
Facilidade para obter respostas de várias partes	2	4	3	4	4	3	5	5	8	13	5	5 Muito bom
Facilidade para encontrar uma solução para o problema	3	5	2	5	5	2	5	5	7	15	8	
Organização dos problemas por grupos												
Estudante												
Facilidade para reportar um problema	3	5	4	5	5	3	4	4	10	14	4	
Satisfação por ver o problema a ser tratado	3	4	3	5	2	2	4	8	8	13	5	
Nível de envolvimento na resolução de problemas	3	3	2	5	5	1	4	6	6	12	6	
Nível de confiança para reportar um problema	2	3	4	2	5	3	2	9	7	7	-2	
Manager												
Facilidade para tratar dos problemas levantados	3	4	3	4	3	3	4	9	9	12	3	
Consciência dos problemas por grupo	3	5	2	5	2	2	5	5	7	15	8	
Facilidade para resolver problemas	2	4	3	4	3	3	4	8	8	14	6	
Autonomia para resolver problemas	2	4	3	4	3	3	4	8	8	12	4	
Nível de confiança para tratar de um problema	3	4	3	3	3	3	4	9	9	11	2	
Administrador												
Mantém-se a par dos problemas atuais												
Ter em atenção algumas situações reportadas em estatísticas												
Facilidade para resolver um problema												
Confiança na resolução de problemas por parte de outros												
Dificuldade para gerir o sistema												

Figura A.1: Resultados obtidos pelo teste comparativo entre sistemas

Resultados obtidos nos testes realizados

	Avaliação					
Estudante	Elemento 1	Elemento 2	Elemento 3	Total		
Dashboard Pessoal	3	5	4	4,0		
Listagem de problemas	5	5	3	4,3		
Reportar um problema	5	5	4	4,7		
Tagar os problemas	4	4	4	4,0		
Pesquisa por nome	4	3	4	3,7	1	Inútil
Pesquisa por tag	3	4	4	3,7	3	Mais valia
					5	Indispensável
Manager						
Listagem de grupos que gere	4	4	3	3,7		
Listagem de problemas	4	5	4	4,3		
Reportar um problema	4	5	4	4,3		
Tagar os problemas	4	4	4	4,0		
Pesquisa por nome	4	3	4	3,7		
Pesquisa por tag	3	4	4	3,7		
Página de um Problema	5	4	4	4,3		
Comentar um problema	5	5	3	4,3		
Administrador						
Semáforo geral	3	4	4	3,7		
Ver problemas mais recentes	4	5	5	4,7		
Configurar semáforo	3	3	3	3,0		
Ver todos os problemas listados	3	4	4	3,7		
Ver todos os utilizadores	5	3	4	4,0		
Validar utilizadores	5	3	5	4,3		
Resolver um problema	5	3	5	4,3		

Figura A.2: Resultados obtidos pelo teste às avaliações do sistema proposto